

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **06-291991**

(43)Date of publication of application : **18.10.1994**

(51)Int.Cl.

H04N 1/393

H04N 1/32

H04N 1/411

(21)Application number : **04-102622**

(71)Applicant : **MATSUSHITA GRAPHIC
COMMUN SYST INC**

(22)Date of filing : **22.04.1992**

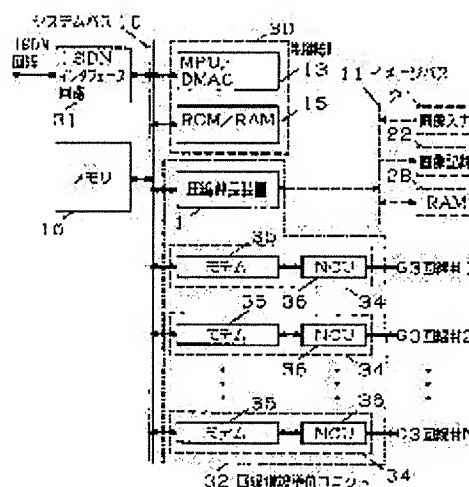
(72)Inventor : **HORIE HITOSHI
OZAKI TORU
SHIRAI HIDEYUKI**

(54) PICTURE COMMUNICATIONS EQUIPMENT

(57)Abstract:

PURPOSE: To realize a simplified facsimile mail equipment and the reduction in the cost, to facilitate line extension and to reduce the cost thereof.

CONSTITUTION: A compander 1 whose speed is ultra-high is employed and shares code conversion processing by the portion of N lines. A unit 32 for line extension is defined as the combination of one compander 1 with the N pieces of communications units 34. Data are sent simultaneously to N pieces of G3 facsimile reception terminal equipments. Reception capability is recognized in the facsimile procedure and the code conversion accordingly is executed by the compander 1 at a high speed.



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-291991

(43)公開日 平成6年(1994)10月18日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 1/393		4226-5C		
1/32	E	7232-5C		
1/411		9070-5C		

審査請求 有 請求項の数 3 O L (全 43 頁)

(21)出願番号 特願平4-102622

(22)出願日 平成4年(1992)4月22日

(71)出願人 000187736

松下電送株式会社

東京都目黒区下目黒2丁目3番8号

(72)発明者 堀江 等

東京都目黒区下目黒2丁目3番8号 松下電送株式会社内

(72)発明者 尾崎 透

東京都目黒区下目黒2丁目3番8号 松下電送株式会社内

(72)発明者 白井 秀行

東京都目黒区下目黒2丁目3番8号 松下電送株式会社内

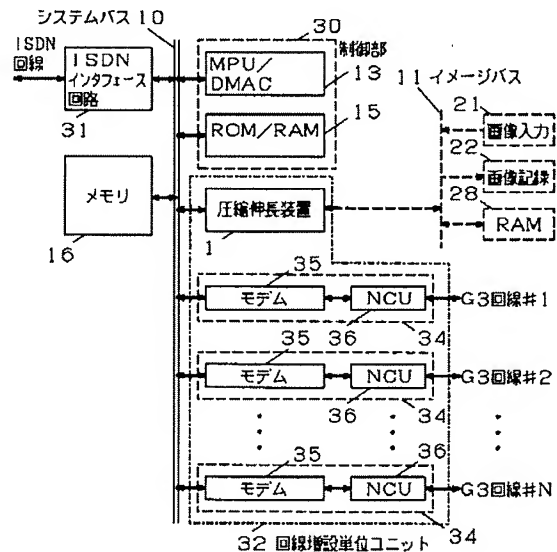
(74)代理人 弁理士 小銀治 明 (外2名)

(54)【発明の名称】 画像通信装置

(57)【要約】

【目的】 ファクシミリメール装置の単純化、コスト削減を実現する。回線増設を容易化、低コスト化する。

【構成】 超高速の圧縮伸長装置1を用い、それ1個でN回線分の符号変換処理を担う。1個の圧縮伸長装置1とN個の通信ユニット34の組み合わせを回線増設の単位ユニット32とする。N個のG3ファクシミリ受信端末ヘデータを同時に送信する。ファクシミリ手順中で、受信能力を認識し、それに応じた符号変換を圧縮伸長装置1で高速に実行する。



【特許請求の範囲】

【請求項1】 複数の回線を用い、同時に複数の画像受信端末に対し画像符号データを送信するための通信手順を実行し、通信手順中に各画像受信端末の受信能力に応じた画像符号データの符号変換処理を実行する画像通信装置であって、回線インターフェイスのための回線対応の複数の通信ユニットと、複数回線分の画像受信端末に対する符号変換処理に共通に用いられる符号変換装置と、複数の画像受信端末に対する発呼及び通信手順の制御並びに前記符号変換装置による符号変換処理の制御を行なう制御部と、前記符号変換装置による符号変換処理の前後の画像符号データを一時的に記憶するためのメモリとをバス接続してなる画像通信装置。

【請求項2】 N個の通信ユニットと1個の符号変換装置との組み合わせが回線増設の単位ユニットを構成することを特徴とする請求項1記載の画像通信装置。

【請求項3】 複数の回線を用い、同時に複数の画像受信端末に対し画像符号データを送信するための通信手順を実行し、通信手順中に各画像受信端末の受信能力に応じた画像符号データの符号変換処理を実行する画像通信装置であって、回線インターフェイスのための回線対応の複数の通信ユニットと、複数回線分の画像受信端末に対する符号変換処理に共通に用いられる符号変換装置と、複数の画像受信端末に対する発呼及び通信手順の制御並びに前記符号変換装置による符号変換処理の制御を行なう制御部と、前記符号変換装置による符号変換処理の前後の画像符号データを一時的に記憶するためのメモリとをバスで接続してなり、前記符号変換装置は、画像符号データの符号変換処理のための複数の処理ブロックと、この複数の処理ブロックの処理実行に必要な複数のラインメモリと、外部の前記バスとのインターフェイスのためのブロックと、前記ラインメモリ及び前記各ブロックに対するデータ転送のための内部バスと、前記各ブロックを制御するブロックとを具備する画像通信装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数の回線を用い同時に複数の画像受信端末に対し画像符号データを送信するための通信手順を実行し、通信手順中に、各画像受信端末の受信能力を認識し、それぞれの受信能力に応じた画像符号データの符号変換を行なう画像通信装置に関する。

【0002】

【従来の技術】 従来、この種の画像通信装置として所謂ファクシミリメール装置があり、これは図36に示すような構成であった。ここに示す例は、ISDN回線によってG4ファクシミリ送信端末からファクシミリデータ（画像符号データ）を受信し、このデータをG3回線（#1～#N）を用い、同時に最大N個のG3ファクシミリ受信端末へ送信するもので、ISDNインターフェ

ース回路50、メモリ51、制御部52、G3回線の収容本数と同数のG3回線対応ブロック53をシステムバス54で接続してなる。

【0003】 各G3回線対応ブロック53は、モデム54とNCU（網制御回路）55、圧縮伸長装置56、拡大縮小装置57を含み、圧縮伸長装置56、拡大縮小装置57及びメモリ58をイメージバス59により接続してなるものである。制御部52はファクシミリ手順制御、各部の動作制御、データ転送制御等を行なうもので、マイクロプロセッサ/DMAコントローラユニット60、メモリ（ROM/RAM）61等からなる。

【0004】 G4ファクシミリ送信端末からの受信データはISDNインターフェイス回路50を経由してメモリ51に蓄えられる。この受信の終了後、制御部52はG3回線（#1～#N）と1対1に対応したG3回線対応ブロック53を介してN個のG3ファクシミリ受信端末へ発呼し、ファクシミリ手順中で各ファクシミリ受信端末の受信能力（符号化方式、受信紙サイズ）を認識し、メモリ51に蓄えられているデータをファクシミリ受信端末の受信能力に応じたデータに符号変換する処理（紙サイズ変換・画像変換を含む）を該当のG3回線対応ブロック53によって行なわせ、変換データをファクシミリ受信端末へ送信する。

【0005】 この場合の通信シーケンス例を図37に示す。この例では、各G3回線に接続されたG3ファクシミリ受信端末の受信能力を、ファクシミリ手順のフェーズBで受信したDIS信号より認識し、その時点から符号変換処理（受信データの復号化—拡大縮小—符号化）を開始し、ファクシミリ手順のフェーズCに入る前に符号変換処理を終了する。ただし、符号変換処理をフェーズC中に実行させてもよい。

【0006】

【発明が解決しようとする課題】 しかし、かかる構成によれば、高価なG3回線対応ブロック53を、同時に処理する必要があるG3回線数Nと同数設ける必要があるため、収容回線数が多いと装置構成が著しく複雑高価になるという問題があった。

【0007】 このような問題の背景としては、G3回線対応ブロック53による符号変換速度が遅かったことがある。すなわちG3回線対応ブロック53では、受信データを圧縮伸長装置56へ送って復号化し、復元された画像データを外部のメモリ58へ一旦格納し、このメモリ58内の画像データに対して拡大縮小装置57によって拡大縮小（画像変換）を行ない、拡大縮小後の画像データをメモリ58から圧縮伸長装置56へ送って符号化し、符号化データをメモリ51へ戻すというシリアルな処理系列によって符号変換が行なわれる。また、圧縮伸長装置56による符号化または復号化処理の際の参照ラインの画像データは、外部のメモリ58上のラインメモリに置かれる。したがって、符号変換処理に伴うイメー

ジバス59のアクセス回数が非常に多く、そのアクセス時間が長いために処理の高速化に限界があったのである。

【0008】本発明は、上述の問題点に鑑みてなされたもので、同時に処理しなければならない回線数が多い場合にも構成が単純で安価なファクシミリメール装置等を提供することと、回線の増設に容易に対応でき、かつそれに伴うコスト上昇が少ないファクシミリメール装置等を提供することとを目的とする。

【0009】

【課題を解決するための手段】本発明は上述の課題を解決するため、複数の回線を用い、同時に複数の画像受信端末に対し画像符号データを送信するための通信手順を実行し、通信手順中に各画像受信端末の受信能力に応じた画像符号データの符号変換処理を実行する画像通信装置において、高速の符号変換装置を用い、回線インターフェイスのための回線対応の通信ユニットと符号変換装置とを独立させ、1個の符号変換装置を、同時に通信手順が実行される複数の画像受信端末に対する符号変換処理に共通に用いる。

【0010】また本発明は、かかる画像通信装置の構成において、N個の回線ユニットと1個の符号変換装置との組み合わせを回線増設の単位ユニットとする。

【0011】また本発明は、かかる画像通信装置の構成において、画像符号データの符号変換処理のための複数の処理ブロックと、この複数の処理ブロックの処理実行に必要な複数のラインメモリと、外部の前記バスとのインターフェイスのためのブロックと、前記ラインメモリ及び前記各ブロックに対するデータ転送のための内部バスと、前記各ブロックを制御するブロックとを具備する改良された符号変換装置を用いる。

【0012】

【作用】本発明は上述の構成によって、符号変換装置の個数を収容回線数より少なくでき、収容回線数と同数の通信ユニットが必要であることは従来装置でも同様であるから、収容回線数が多い場合でも画像通信装置の全体的構成を大幅に単純化できる。そして、符号変換装置は通信ユニットに比べ相当に高価なものにならざるを得ないので、符号変換装置の減少によって画像通信装置のコストを大幅に削減できる。

【0013】また、1個の符号変換装置とN個の通信ユニットからなる単位ユニットの追加のみによって、容易にN回線単位の回線増設を行なうことができる。そして、追加した単位ユニット分のコスト上昇以外に格別のコスト上昇要因を伴わないので、回線増設コストも少なく済む。

【0014】また、本発明の画像通信装置に用いられる改良された符号変換装置の構成によれば、変換前の画像符号データの外部入力と変換後の画像符号データの外部出力のために外部バスをアクセスするのみで、それ以外

に外部バスをアクセスすることなく符号変換処理を実行でき、また処理実行に必要なラインメモリも内部にあるので、それに対するアクセスも高速に行なうことができるため、極めて高速の符号変換処理が可能である。その結果、1個の符号変換装置で多数回線分の符号変換処理を担わせることによって、高価な符号変換装置の個数を大幅に減らすことができる。

【0015】

【実施例】図1は、本発明の一実施例によるファクシミリメール装置の概略ブロック図である。ここに示す例は、ISDN回線よりG4ファクシミリ送信端末からのファクシミリデータを受信し、このデータを符号変換して同時に最大N個のG3ファクシミリ受信端末へ送信する。

【0016】図1において、1は極めて高速の符号変換処理が可能な改良された圧縮伸長装置である。この圧縮伸長装置1は、後に詳細に説明するように、符号変換装置としてだけでなく、圧縮装置、伸長装置または画像変換装置として用いることができるものである。16は画像符号データ（受信データまたは送信データ等）を一時的に蓄積するためのメモリである。31はISDN回線とのインターフェイスのためのISDNインターフェース回路である。

【0017】34はG3回線（#1～#N）とのインターフェイスのための回線対応の通信ユニットであり、この例ではN個ある。各通信ユニット34はモデム35とNCU35からなる。

【0018】30は制御部であり、ファクシミリ手順制御、各部の動作制御、データ転送制御等を行なうものである。この制御部30はプログラム制御方式のものである。DMAコントローラを含むマイクロプロセッサユニット（MPU）13、プログラムやデータを格納するメモリ（ROM/RAM）15、その他MPU周辺回路（不図示）からなる。10は各部相互間のデータや制御情報の伝送路としてのシステムバスである。

【0019】圧縮伸長装置1は1台でN回線分の符号変換処理を担う。1台の圧縮伸長装置1とN個（N回線分）の通信ユニット34の組が1つの回線増設の単位ユニット32を構成しており、この単位ユニット32の追加のみによってN回線単位で回線増設に対応できる。

【0020】図2は、このような増設の例である。この例では、単位ユニット32が2個設けられているため、2N本のG3回線（#1～#N、#N+1～#2N）により2N台のG3ファクシミリ受信端末への同時送信が可能である。他の構成は何等変更を必要としない。なお、1個の単位ユニット32に実際に接続される回線数をN本より少なくすることも別段差し支えない。

【0021】このファクシミリメール装置の全体的な動作は以下の通りである。なお、ここでは図1の構成であるとして説明するが、図2のように単位ユニット32が

増設された場合でも同様に考えればよい。

【0022】ISDN回線を介してG4ファクシミリ送信端末から受信された画像符号データは、ISDNインターフェイス回路31を経由してメモリ16に蓄えられる。この際のファクシミリ手順の制御は制御部30によって行なわれる。この手順制御は従来と変わらないので、その詳細説明は省く。

【0023】このG4ファクシミリ送信端末からの受信が終了後、制御部30はG3ファクシミリ受信端末に対する送信処理を開始する。1回線分の送信処理の概略フローは図3(a)に示す通りである。

【0024】制御部30は、G3回線(#1~#N)と1対1に対応した通信ユニット34を介してN個のG3ファクシミリ受信端末へ発呼し(S1)、呼設定がなされたG3ファクシミリ受信端末とのファクシミリ手順を実行し(S2)、その後に関線開放を行なう(S3)。

【0025】制御部30は、各G3ファクシミリ受信端末とのファクシミリ手順において、フェーズBで受信したDIS信号により各G3ファクシミリ受信端末の受信能力を認識する。この認識内容はG3回線(#1~#N)に対応付けてメモリ15に保存されるが、それと同時にメモリ15上のG3回線対応のDIS受信フラグがセットされる。

【0026】また制御部30は、メモリ16内の受信画像符号データに対する各G3ファクシミリ受信端末の受信能力に応じた符号変換処理を、圧縮伸長装置1により実行させる制御を行なう。図3(b)は、この制御の概略フローである。メモリ15上のDIS受信フラグを参照し、フラグがセットされている回線の有無を調べる

(S10)。1回線以上のDIS受信フラグがセットされている場合、その中の1回線を選び、同回線のG3ファクシミリ受信端末の受信能力に応じた符号変換処理を実行させる(S11)。変換された画像符号データは回線別に管理されてメモリ11に格納される。1回線分の符号変換処理を終了すると、同回線に対応したDIS受信フラグをリセットするとともに、メモリ15上の回線対応の符号変換終了フラグをセットする(S12)。全回線分の符号変換処理を終了するまで(S13)、同様の制御を繰り返す。

【0027】このような符号変換処理はファクシミリ手順(S2)のフェーズCに入る前に終了する。逆に言えば、最悪のケース、つまり全回線のG3ファクシミリ受信端末から同時にDIS信号を受信した場合でも、フェーズCに入る前に、全回線分の符号変換を終了できるように、1個の圧縮伸長装置1が受け持つ回線数Nが決定されている。

【0028】その後、制御部30は、各G3ファクシミリ受信端末に対し、そのフェーズCにおいて、メモリ16内の対応した変換符号データを対応の通信ユニット34を介し送信する。

【0029】なお、G3回線(#1~#N)には予め優先順位が決められており、ステップS11において、2回線以上のDIS受信フラグがセットされているときは、その中の優先順位が最も高い1回線を選択する。例えばG3回線(#1~#N)の優先順位が#1>#2>#3>...>#N-1>#N(#1が最高順位)である場合、全回線のDIS受信フラグがセットされているときは、#1から順に選択されることになる。

【0030】なお、ここで説明したフラグを用いた制御や優先制御はあくまで一例であって、他の適当な制御方法を採用してもよいことは当然である。

【0031】図4は通信シーケンスの一例を示している。ここでは、全てのG3回線(#1~#N)から同時にDIS信号を受信したと仮定している。この場合、前述のように予め決められた優先順位に従って符号変換処理が実行される。一般的にはDIS信号の受信時刻は回線毎にばらつきがある。

【0032】G3ファクシミリの手順時間は長く、DIS信号を受信してからフェーズCに入るまで通常は5秒程度かかる。本実施例に用いられる圧縮伸長装置1の符号変換処理(紙サイズ変換も含む)の速度は、後に具体的に説明するように、5秒間に16回線分の符号変換処理を実行できるほど高速である。つまり、最大N=16である。もっとも、この数字は最悪ケース、つまり全回線から同時にDIS信号を受信する場合を想定したものである。実際には前述のようにDIS信号の受信時刻は回線毎にばらつくので、実用的にはさらに多くの回線に対する符号変換処理を1個の圧縮伸長装置1で担うことができる。

【0033】なお、ここまではISDN回線より受信したファクシミリデータを符号変換してG3ファクシミリ受信端末へ送信するものとして説明したが、G3回線より受信したファクシミリデータを同様に送信することも可能である。

【0034】圧縮伸長装置の全体的構成

図5は、本実施例に用いられる改良された圧縮伸長装置1の内部構成を示すブロック図である。この例は、符号化処理、復号化処理、画像変換(紙サイズ変換のための拡大縮小)処理、それらの組み合わせ処理を高速に行なうことができ、その組み合わせ処理の一つが符号変換処理である。

【0035】また、この例は、システムバス10とのインターフェイスが可能であるばかりでなく、図1に破線として示すように、外部のイメージバス11とのインターフェイスも可能である。そして、このイメージバス11上のRAM28を介して、あるいは直接に、画像データ入力部21からのデータ入力、画像記録部22へのデータ出力等の制御を行なうことができる。

【0036】図5において、100は画像バス11とのインタフェース機能を実現するイメージバス制御部であ

る。200はRAMであり、内部処理のためのラインメモリ及びパラメータレジスタとして使用される。このRAM200は、制御部30のMPU13（図1）からもアクセスできる。300は内部データバス（BEデータバス）1700によるRAM200に対するDMA転送を制御する内部バスDMA制御部、400はMPU（図1）とインターフェイスするためのシステムバス制御部である。

【0037】500は種々のレジスタとして利用されるワーキングレジスタであり、実際的にはRAMが用いられる。600～800は16ビット幅データの変化画素アドレスを検出するための変化画素検出部、900～1100は変化画素アドレス情報の一時記憶のためのFIFOバッファ、1200は内部処理の実行に関連して利用される算術論理演算部、1300はMH/MR/MMRの復号器、1400は画像の主走査方向の画像変換（拡大縮小）を行なう画像変換部、1500はMH/MR/MMRの符号器、1600は装置動作の制御のためのマイクロプログラム制御部である。

【0038】なお、変化画素検出部600～800とFIFOバッファ900～1100は、対応した処理ブロック1300～1100に含めることもできる。ただし、この場合、変化画素検出部600とFIFOバッファ900は二つの処理ブロック1300、1500に共用されるため、同じものを組追加する必要がある。

【0039】1700はDMA制御バスであり、各部からのDMA転送要求信号線や各部へのDMA転送許可信号線から成る。内部データバス（BEデータバス）1800は、主に画像データの転送に使われる16ビットのバスである。1900も16ビットの内部データバス（BCデータバス）であり、これは主に符号データの転送に利用される。図5には示されていないが、マイクロプログラム制御部1600と装置内各部との間にマイクロプログラム制御バスが存在する（図9などを参照）。

【0040】符号器の構成

図6は符号器1500のブロック図である。図6において、変化画素アドレス制御部1502は、FIFOバッファ900より参照ラインの変化画素アドレスを、FIFOバッファ1100より符号化ラインの変化画素アドレスをそれぞれ取り込み、順序付けして符合化モード判定部1504に入力する。

【0041】この符号化モード判定部1504は、入力した変化画素アドレス情報より符号化モード（バス、垂直、水平モード）の判定を行なう。符号テーブル検索部1506は、符号化モードの判定結果に基づいて内部の符号テーブルを検索し符号割り当てを行なう。

【0042】パッキング処理部1508は、符号テーブル検索部1506より出力された可変長符号の16ビット/ワードの符号データへの変換（ワードパッキング）を行ない、ワード単位で内部データバス1900または

1800へ出力する。1510は符号器1500の全体的制御のためのメインシーケンサである。内部RAM200とのDMA転送の要求は、このメインシーケンサ1510より出される。1512～1518はメインシーケンサ1510の制御下で対応処理部1502～1508を制御するサブシーケンサである。

【0043】符号器1500はまた、内部データバス（BEデータバス）1800を通じて1ライン幅（1ラインの画像データのワード数）がセットされるレジスタ1520、1ライン符号数（1ラインの符号データのワード数）をカウントするためのカウンタ1522を有する。このカウンタ1522の値は内部データバス1800に出力できる。

【0044】1524はMG3符号化（後述）に関連して設けられた比較器で、レジスタ1520の値とカウンタ1800の値を比較する。この比較出力はマイクロプログラム制御バス1602の状態信号に反映される。マイクロプログラム制御部1600は、マイクロプログラム制御バス1602を介し、符号器1500に対する符号化モードの指定や起動等の制御を行なうことができ、また符号器1500の状態を取得できる。

【0045】復号器の構成

図7は復号器1300のブロック図である。図7において、符号シフト部1302は、内部データバス（BCデータバス）1900より取り込んだ符号データを解説を終了した符号長分だけシフトし、符号解析部1304に常に未解説の符号データを与える。符号解析部1304は、符号データによって内部の復号ROMの検索を行ない、復号符号を描画部1308へ送る。ただし、MG3符号化の拡張符号を検出した場合、入力した画像データが符号シフト部1302から画像データへ描画部解析部1308へ転送される。

【0046】ここでMG3符号化とは、1ラインの画像データをファクシミリ標準符号データ（MH、MRまたはMMR符号）に符号化し、1ラインの符号データの長さが元の画像データの長さ（または画像データと拡張符号の合計長）を超えた場合に、元の画像データに拡張符号（十数ビット）を付加したデータを符号化出力とする符号化方式である。このMG3符号化方式は、特願平3-2669号の明細書及び図面に具体的に説明されている。

【0047】a0アドレス演算部1306は、FIFOバッファ900より入力した参照ラインの変化画素アドレス情報と符号解析部1304から入力した復号符号とから、符号化ラインの起点もしくは基準変化画素a0のアドレス（CITT勧告T、4参照）を計算する。描画部はa0アドレスと白/黒情報から画像データを生成し、生成した画像データをワード（16ビット）単位で内部データバス（BEデータバス）1800へ出力する。

【0048】1310は復号器1300の全体的制御を行なうメインシーケンサ、1312～1318はメインシーケンサ1310の制御下で対応機能ブロック1302～1318を制御するサブシーケンサである。DMA転送要求はメインシーケンサ1310より出る。

【0049】復号器1300はまた、復元された画像データより白データ（全ビットが白ビットのワード）を検出するための比較器1320と、連続EOL数と1ライン幅（符号数）が内部データバス（BEデータバス）1800よりセットされるレジスタ1322、1324を有する。比較器1320による比較結果に基づきメインシーケンサは白ライン（全ビットが白画素のライン）の判定を行ない、その判定結果を状態信号としてマイクロプログラム制御バス1602に出力する。

【0050】また、メインシーケンサ1310によって1ライン毎に復号エラーがチェックされる。このチェック結果は状態信号として出力される。マイクロプログラム制御部1600は、マイクロプログラム制御バス1602を介し、復号器1300に対して復号化モードの指定、起動などの制御と状態監視を行なうことができる。

【0051】画像変換部の構成

図8は画像変換部1400のブロック図である。図8において、レジスタ1402はFIFOバッファ1000より入力する変化画素アドレス（14ビット）と色情報（B/W）を保持するもので、レジスタ1404は内部データバス（BEデータバス）1800を通じて拡大縮小率をセットされるものである。乗算器1406は、その変化画素アドレスと拡大縮小率を乗算することによって、拡大縮小後の変化画素アドレスを求め描画部1408に与える。

【0052】描画部1408は、与えられた変化画素アドレスとレジスタ1402より与えられる色情報に基づき拡大縮小後の画像データを生成する。この画像データはレジスタ1410を介しワード単位で内部データバス（BEデータバス）1800へ出力される。1412は内部データバス1800を介し変換前の1ライン幅（ワード数）をセットされるレジスタ、1414は変換後の1ライン幅（ワード数）をカウントするためのレジスタである。1416は画像変換部1400内各部を制御するシーケンサであり、DMA転送要求も出す。

【0053】算術論理演算部、ワーキングレジスタなどの構成

図9は算術論理演算部1200及びワーキングレジスタ（RAM）500とその周辺の構成、並びに他の機能ブロックとの接続構成を示している。図9において、1202は算術論理演算部1200の中心をなす16ビットALU（シフト含む）である。

【0054】図から明らかなように、RAM200など

からのデータをALU1202にロードして必要な演算を行ない、演算結果をRAM200などに書き込むことができる。また、ワーキングレジスタ（RAM）500上のレジスタの操作やチェックをALU1202を経由して行なうことができる。

【0055】図9において、1204と1206はALU1202の入力レジスタ、1207と1207はALU1202の入力選択のためのセクタ、1210は算術論理演算部1200のローカルバス、1211はローカルバス1210への出力バッファである。1212と1213はローカルバス1210と内部データバス（BEデータバス）1800との間のデータ転送のためのバッファである。

【0056】1214はマイクロプログラム制御バス1602上の周辺アドレスをデコードしALU1202周辺の制御信号を出すデコーダ、1216はワーキングレジスタ500のリード/ライトを制御するR/W制御回路（デコーダ）、1218はマイクロプログラム制御部1600によって制御されるアドレスポインタ（カウンタ）、1220はアドレスポインタ1218の値またはマイクロプログラム制御バス1602より与えられるアドレスを選択しアドレスバス1220へ出すセクタである。

【0057】マイクロプログラム制御部、システムバス制御部の構成

図10はマイクロプログラム制御部1600とシステムバス制御部400の構成の説明図である。

【0058】この圧縮伸長装置1は符号化、復号化の処理チャンネルを2チャンネル有し、ライン単位にチャンネルを切り替えて処理を実行することができる。このような処理実行を容易にするため、システムバス制御部400にチャンネル0（CH0）用のレジスタセット402と、チャンネル1（CH1）用のレジスタセット404がある。またシステムバス制御部400には、システムバスタイミング制御部406と、図5に示されるように、データバッファ408、DMAコントローラ410、クロック発生器412なども含まれる。

【0059】マイクロプログラム制御部1600は一般的な構成であって、様々なコマンドの処理のためのマイクロプログラムを格納したマイクロROM1601のほか、マイクロプログラム実行制御のためのプログラムカウンタ1603、スタック1604、スタックポインタ1605、命令レジスタ1606、命令デコーダ1607を含む。

【0060】マイクロプログラム制御部1600はさらに、各マクロコマンドのためのマイクロプログラムのスタートアドレスを格納したマクロROM1608、レジスタセット402、404中のコマンドレジスタにセットされたマクロコマンドをマクロROM1608へ入力するためのセクタ1609、マイクロROM1602

の入力切り替えのためのマルチプレクサ1610、マイクロプログラム制御バス1602上の状態信号及びシステムバス制御部400からの起動信号をマルチプレクサ1610へ制御信号として入力するためのマルチプレクサ1611などを含む。

【0061】内部RAMの使用法

図11はRAM200の使用法の説明図である。RAM200のリニアなアドレス空間は、チャンネル0（CH0）用パラメータレジスタセットの領域201、チャンネル1（CH1）用のパラメータレジスタセットの領域204、画像メモリ領域206に分割されて使用される。各チャンネルのパラメータレジスタセット領域202、204は、符号化コマンド、復号化コマンド、その他コマンド及びDMAのためのパラメータレジスタ領域208～214に分割される。

【0062】画像メモリ領域206は、複数のラインメモリの領域に分割され、分割領域が後に図13ないし図15を参照して説明するように処理内容に応じた各種のラインメモリとして使用される。

【0063】イメージバス制御部の構成

図12はイメージバス制御部100のブロック図である。イメージバス制御部100は、画像データのDMA転送のためのDMAコントローラ102、アドレスカウンタ104及びデータバッファ106を含む。このイメージバス制御部100の制御によって、次の4種類の画像データのDMA転送が可能である。

- a) イメージバス上のI/Oデバイス（画像入力部21）からメモリ（RAM28）への転送
- b) イメージバス上のメモリ（RAM28）からI/Oデバイス（画像記録部22）への転送
- c) イメージバス上のメモリ（RAM28）から圧縮伸長装置1への転送
- d) 圧縮伸長装置1からイメージバス上のメモリ（RAM28）への転送

各DMA転送に対応して、アドレスカウンタ104は4組のアドレスレジスタ110とインクリメント112より構成されている。同様に、DMAコントローラ102には、4組の転送数レジスタ116とデクリメント116が含まれる。DMAコントローラ102にはまた、DMA要求の優先制御のための優先コントロール118やタイミング制御部120などが含まれる。

【0064】ラインメモリ関連の構成

図13、図14及び図15は、内部バスDMA制御部300の内部構成、マイクロプログラムによってワーキングレジスタ500上に定義されるアドレスレジスタ502及びRAM200の画像メモリ領域206上に定義されるラインメモリ216の内訳と、それらの対応関係を説明するための図である。

【0065】図13図は符号化コマンド処理の場合を示し、図14は復号化コマンド処理の場合を示し、図15

は符号変換コマンド処理の場合を示している。以下の説明において、アドレスレジスタ502及びラインメモリ216の個々を区別するために必要な場合には図13、図14または図15に示された名称INPUT～DIRを用いる。

【0066】内部バスDMA制御部300には、ラインメモリ216と同数のアドレスカウンタ（A～J）302、RAM200と符号器1500、復号器1300などの処理ブロックとの間のDMA転送を制御するDMA制御部304、アドレスカウンタ302を選択するためのセクタ306などが含まれる。

【0067】RAM200上のラインメモリ216の領域、アドレスカウンタ302及びアドレスレジスタ502は1対1に対応している。符号化処理と復号化処理のための参照ライン用ラインメモリは2チャンネル分あるため、外部のMPUは、画像処理装置1に符号器1500、復号器1300が2個あるように動作させることが容易である。

【0068】図16は、ワーキングレジスタ500上に定義されるアドレスレジスタ502の構造を示す。INPUTアドレスレジスタが例として示されているが、他のアドレスレジスタ502の構造も同様である。

【0069】図示のように、アドレスレジスタ502の下位11ビットはラインメモリの先頭アドレスである。上位の4ビット（A～E）はフラグビットで、その意味は次のとおりである。

A：“1”のとき対応ラインメモリに有効データがあることを示す。

B：“1”のとき対応ラインメモリの内容が縮小対象データであることを示す。

C：“1”のとき対応ラインメモリの内容が最終ラインデータであることを示す。D：レジスタによって意味が異なる。

E：“1”のとき対応ラインメモリの内容が拡大対象データであることを示す。

【0070】マイクロプログラムは、これらのフラグビットの操作やチェックを算術論理演算部1200を用いて行なうことができる。

【0071】圧縮伸長装置の動作

以上のように構成された圧縮伸長装置1の各処理の動作について、図1に示したシステム構成と関連させて説明する。

【0072】圧縮伸長装置1の画像データの入出力経路は次の通りである。

- a) 画像入力部21→圧縮伸長装置1
- b) 画像入力部21→RAM28→圧縮伸長装置1
- c) 圧縮伸長装置1→画像記録部22
- d) 圧縮伸長装置1→RAM28→画像記録部22

圧縮伸長装置1のイメージバス制御部100は、このような画像データのDMA転送をサポートしているが、画

像入力部1からRAM28に転送するのがDMAチャンネル0、RAM28から画像記録部22に転送するのがDMAチャンネル1である。

【0073】圧縮動作の説明

(概要) 制御部30のMPU13(図1)は、圧縮伸長装置1に対してマクロコマンドを発行することによって動作の指示を与える。MPU13はまず、システムバス制御部400内の各種レジスタの設定を行なう。この中には、符号化チャンネルCH0、CH1の指定も含まれる。

【0074】このレジスタ設定の終了後、システムバス制御部400内の指定チャンネルのコマンドレジスタ402Aまたは404B(図10)に符号化コマンドを書き込む。このコマンドはセクタ1609を通してマクロROM1608で解読され、符号化プログラムのスタートアドレスが出力される。このアドレスからマイクロROM1601内の符号化プログラムが実行される。圧縮伸長装置1内の各処理ブロックは、マイクロROM1601に書かれたプログラムで制御される。

【0075】既に説明したように、符号化コマンド処理の場合に定義されるラインメモリ216とアドレスレジスタ502は図13に示すとおりである。各ラインメモリの内容または役割は次のとおりである。

INPUT : 入力ラインの画像データ(入力バッファ)
CONVR : 主走査変換前ラインの画像データ
CONVW : 主走査変換後ラインの画像データ
CODING : 符号化ラインの画像データ
BC1 : 符号データ(出力バッファ)
BC : 符号データ(出力バッファ)
COR : 符号化チャンネル0用の参照ラインの画像データ
C1R : 符号化チャンネル1用の参照ラインの画像データ
DOR : 復号化チャンネル0用の参照ラインの画像データ
D1R : 復号化チャンネル1用の参照ラインの画像データ

(図19に沿った説明) 図19に符号化プログラムのフローの一例を簡略化して示す。このフローに沿って圧縮動作を詳細に説明する。

【0076】符号化コマンドが発行されると、処理2001で必要なパラメータをRAM200のパラメータレジスタセット領域202(CH0)または204(CH1)からワーキングレジスタ500にロードする。アドレスレジスタ502には同名のラインメモリ216の領域の先頭アドレスが設定される。

【0077】処理2002、処理2003で、イメージバス11上のDMA転送要求(画像入力部21からRAM28への転送要求、RAM28から画像記録部22へ

の転送要求)があった時に、DMA転送処理を行なう。マイクロプログラムは、DMA転送があれば、イメージバス制御部100のスタートアドレスレジスタ110にスタートアドレスを、転送数レジスタ114に転送ワード数を、それぞれ設定し、起動をかけ、起動フラグを”1”にセットする(図12)。この後、イメージバス制御部100がDMA転送を実行する。

【0078】次の処理2004は、イメージバス11上のRAM(ラインバッファ)28より、圧縮伸長装置1のRAM200上のINPUTラインメモリに1ラインの画像データを入力する処理である。

【0079】(図20に沿った説明: 画像データ入力) この画像データ入力処理のフローを図20に示す。図20において、マイクロプログラムは処理2101、2102で、イメージバス制御部100が動作中でなく、かつイメージバス制御部100の起動フラグがリセット状態であることを確認する。これを確認できた場合、処理2103で、内部バスDMA制御部300内のアドレスカウンタA(図13)に、INPUTアドレスレジスタよりINPUTラインメモリの先頭アドレスを内部データバス1800経由で設定する。処理2104で、イメージバス制御部100のアドレスレジスタ104(図12)の一つに外部RAM28のアドレスを設定する。

【0080】ここでは、画像入力部21→RAM28→圧縮伸長装置1の経路で画像データを入力する場合を想定している。

【0081】処理2105で、イメージバス制御部100内の転送数レジスタ114(図12)の一つに1ラインのワード数を設定する。処理1706でイメージバス制御部100をメモリリード・モードに設定し、処理2107で起動し、処理2108で起動フラグを”1”にセットする。

【0082】起動後は、1ワードのデータをイメージバス制御部100がリードする毎にインクリメント112はメモリリードアドレスをインクリメントし、デクリメント116は転送ワード数をデクリメントする。

【0083】イメージバス制御部100に読み込まれた画像データは、内部データバス(BEバス)1800経由でRAM200上のINPUTラインメモリに転送される。この転送は、イメージバス制御部100がRAM200へのDMA転送要求を出し、内部バスDMA制御部300内のDMA制御部304が内部データバス1800の制御権をイメージバス制御部100に与えることで実行される。1ワードの画像データをINPUTラインメモリに転送すると、内部バスDMA制御部300内のアドレスカウンタAもインクリメントされる。

【0084】イメージバス制御部100に設定した転送ワード数がゼロになるまで、以上の動作を繰り返す。転送中は処理2101の判断ですぐにリターンする。

【0085】1ラインの転送が終了すると、INPUT

ラインメモリに1ライン分の画像データが入る。処理2109以降は1ライン入力後の処理である。

【0086】処理2109では、処理2108でセットした起動フラグをリセットする。処理2110で、次のラインが入っている外部RAM28の先頭アドレスを計算する。処理2111では、符号化コマンドで連続して処理するライン数をデクリメントし、残りの処理ライン数を計算する。処理2112で、処理2111の結果より直前に入力したラインが最終ラインかどうかを判断する。最終ラインのときは、処理2113でINPUTアドレスレジスタのCフラグを”1”にセットする。処理2114でINPUTアドレスレジスタのAフラグを”1”にセットする。このAフラグやCフラグの状態は、後述するようにアドレスレジスタの内容を交換する過程で後続の処理に引き渡される。

【0087】図17は、INPUTラインメモリに1ライン分のデータが入った時の状態を示したものである。ここでは、INPUTラインメモリはアドレスXXXより始まるメモリ領域216Aであると仮定している。画像データが入り終わると、INPUTアドレスレジスタのAフラグは”1”にセットされ、INPUTラインメモリに有効データが有ることが分かる。CONVRアドレスレジスタはYYYを示し、そのAフラグは”0”であるから、CONVRラインメモリがアドレスYYYより始まるメモリ領域216Bで、これが空いていることが分かる。

(図19に沿った説明の続き) 図19のフローに戻る。マイクロプログラムは、処理2005でINPUTアドレスレジスタのA=1のチェックをし、A=1ならば処理2006でCONVRアドレスレジスタのA=0のチェックを行なう。

【0088】INPUTのA=1かつCONVRのA=0ならば、すなわちINPUTラインメモリに有効データがあって、CONVRラインメモリが空いている場合、処理2007でINPUTアドレスレジスタとCONVRアドレスレジスタの内容を交換する。

【0089】(ラインメモリ間のデータ受け渡し) このラインメモリ間のデータの受け渡しを図17と図18図によって説明する。図17は処理2007の実行前の状態である。図18はアドレスレジスタINPUT、CONVRの内容を入れ換えた後の状態である。図18において、CONVRアドレスレジスタはメモリ領域216Aの先頭アドレスXXXを指し、INPUTアドレスレジスタはメモリ領域216Bの先頭アドレスYYYを指す。これは実質上、INPUTラインメモリに入力されたデータがCONVRラインメモリに渡され、INPUTラインメモリに空き領域が渡されたということである。

【0090】このように、RAM200上での実際のデータ移動を伴わない手法であるため、ラインメモリ間の

データ受け渡しは瞬時に行なわれる。

【0091】(図19に沿った説明の続き) 次の処理2008は、主走査方向の拡大縮小(画像変換)処理である。この処理では、CONVRラインメモリのデータを変換してCONVWラインメモリに書き込む。

【0092】この処理2008において、マイクロプログラムは画像変換部1400を起動する前に次の設定を行なう。内部バスDMA制御部300内のアドレスカウンタB、Cに、アドレスレジスタCONVR、CONVWに設定されている先頭アドレスをロードする(図13)。制御部300のMPU13(図1)によってRAM200内のパラメータレジスタ208に設定された拡大縮小率を、画像変換部1400内のレジスタ1404

(図8)に設定する。CONVRラインのワード数をレジスタ1412に設定する。このような初期設定後、マイクロプログラムは画像変換部1400に起動をかけ、当該処理2008を抜ける。

【0093】CONVRラインメモリの画像データは変化画素検出部700へDMA転送されて変化画素データに変換され、そのアドレス情報がレジスタ1402に入力する。乗算器1406によってレジスタ1402、1404の内容が乗算され、変換後の変化画素アドレスデータが得られる。このデータとレジスタ1402内の色情報をもとに描画部1408で変換後の画像データが作られる。得られた変換後画像データは、CONVWラインメモリにDMA転送される。この場合のCONVRラインメモリからのDMA転送と、CONVWラインメモリへのDMA転送の方法は、イメージバス制御部100からRAM200へのDMA転送と同様である。1ラインの変換が終了すると、CONVWアドレスレジスタのAフラグを”1”にセットする。

【0094】副走査方向縮小が必要な場合、1ライン主走査変換に引き続きCONVWラインメモリ上のラインデータが間引きラインであるか否かの判定とフラグ制御が行なわれる。この処理の内容については後述する。間引きラインと判定された場合、CONVWアドレスレジスタのAフラグはセットされない。

【0095】処理2009、2010で、変換後の有効データの有無と符号化用のCODINGラインメモリが空いているかどうか判断する。CONVWアドレスレジスタのA=1かつCODINGアドレスレジスタのA=0のときには、次の処理2011でアドレスレジスタCONVW、CODINGの内容を交換し、交換後の先頭アドレスを対応したアドレスカウンタ302にロードすることによって、ラインメモリCONVW、CODING間のデータの受け渡しを行なう。間引きラインの場合は、この交換は行なわれない。

【0096】処理2012はCODINGラインメモリのデータの符号処理である。マイクロプログラムは、符号器1500が動作中でなければ、符号器1500に対

し符号化モード（MH、MR、MMR、MG3）を設定し、またレジスタ1520に1ライン幅を設定し、起動をかける。ただし間引きラインの場合、CONVWアドレスレジスタのAフラグは“0”であるので、符号器1500を起動しない。

【0097】起動された符号器1500は、CODINGラインメモリの画像データの符号化処理を、符号化参照ラインメモリC0R（CH0）、C1R（CH1）のいずれかのデータを参照して行ない、その結果を符号データメモリBC1に書き込む。CODINGラインメモリからのデータ読み出し、BC1ラインメモリへのデータ書き込みは内部バスDMA制御部300を通して行なわれる。内部バスDMA制御部300の動きはイメージバス制御部100からRAM200へのデータ転送の場合と同様である。

【0098】符号化ラインの変化画素アドレスは変化画素検出部800によって検出され、参照ラインの変化画素アドレスは変化画素検出部600により検出される。

【0099】1ラインの符号化が終了すると、BC1アドレスレジスタのAフラグは“1”にセットされる。CODINGアドレスレジスタとアドレスカウンタC0R（CH0）またはC1R（CH1）の内容交換によって、参照ラインが更新される。

【0100】マイクロプログラムは、処理2013、2014で符号化の終了とBCラインメモリの空きを確認すると、処理2015でアドレスレジスタBC1、BCの内容を交換することによって、ラインメモリBC1、BCのデータの受け渡しを行なう。

【0101】処理2016では、BCラインメモリからシステムバス10に符号データをDMA転送によって出力させる。この際に、出力する符号量を知る必要があるが、符号化終了時に符号器1500内のカウンタ1522（図6）の内容を参照することで符号量が分かる。

【0102】処理2017では設定ライン数の符号化が終了したかどうかを判断し、終了していなければ処理2002に戻る。符号化が終了していれば、処理2018で符号データが全て外部に出るのを待って符号化コマンドの処理を終了する。

【0103】符号化コマンドの終了は、Cフラグが“1”に設定された最終ラインの符号化が済んだかどうかで判断する。Cフラグの状態は次のようにアドレスレジスタを伝搬する。

INPUT, C=1 → CONVR, C=1

CONVRのC=1ならば画像変換終了後 CONVR, C=1 → CODING, C=1

（圧縮動作のまとめ）以上に説明したように、内部RAM200上のラインメモリにデータが詰まった後は、画像データの入力処理（処理2004）、画像変換処理（処理2008）、符号化処理（処理2012）、符号データ出力処理（処理2016）が並列に動作する。更

に、これらと並行して、イメージバス側のDMA転送（処理2002、処理2003）もできる。

【0104】従って、本圧縮伸長装置1の圧縮処理時間は近似的に次式によって表わすことができる。

処理時間 = max {画像入力時間, 画像変換時間, 符号化時間, 符号出力時間}

図21は符号化コマンド処理におけるラインメモリの使い方を示している。この図からわかるように、ラインメモリINPUT, CONVRをトグルで使用し、ラインメモリCONVW, CODING, C0R/C1Rを循環して使用し、ラインメモリBC1, BCをトグルで使用する。

【0105】なお、主走査変換を行わない場合、図21に付記されているように、CONVWラインメモリのデータは直接的にCODINGラインメモリに渡される。

【0106】また、以上の説明では画像データをイメージバス11側から入力したが、図21に示されるように、本圧縮伸長装置1は符号化すべきデータをシステムバス10からも入力できる。同様に、以上の説明では符号化データをRAM200を経由させシステムバス10に出力したが、符号器1500より直接的にシステムバス10に出力することも可能である。

【0107】伸長動作の説明

（概要）制御部30のMPU13はまず、復号化コマンド処理に関する各種レジスタの設定を行なう。この中には、復号化チャンネル0, 1（CH0, CH1）の指定も含まれる。

【0108】このレジスタ設定の終了後、MPU13はシステムバス制御部400内のコマンドレジスタ402Aまたは404Aに復号化コマンドを書き込む。このコマンドは、マクロROM1608で解読され、復号化プログラムのスタートアドレスが出力される。このアドレスからマイクロROM1601内の復号化プログラムが実行される。

【0109】既に説明したように、符号化コマンド処理の場合に定義されるラインメモリ216とアドレスレジスタ502は図14に示すとおりである。各ラインメモリの内容または役割は次のとおりである。

DECODE : 復元ラインの画像データ

D0R : 復号化チャンネル0用参照ラインの画像データ

D1R : 復号化チャンネル1用参照ラインの画像データ

CONVR : 変換前ラインの画像データ

CONVW : 変換後ラインの画像データ

OUT2 : 出力ラインバッファ

OUT1 : 出力ラインバッファ

OUT : 出力ラインバッファ

符号データはラインメモリD0R（CH0）またはD1

R (CH1) のデータを参照して復号され、復元されたデータはDECODEラインメモリに展開される。1ラインの復号が終了すると、DECODEラインメモリの内容はラインメモリD0RまたはD1Rに渡され、次のラインの復号の際に参照される。それと同時にラインメモリD0RまたはD1Rの内容はラインメモリCONVRに渡されて、画像変換の対象となる。画像変換はCONVRラインメモリのデータに対して行なわれ、変換後の画像データはCONVWラインメモリに書き込まれる。

【0110】CONVWラインメモリ内の変換後データは、OUT2ラインメモリが空いていれば、すぐにOUT2ラインメモリに渡される。OUT2ラインメモリのデータは、OUT1ラインメモリが空いていれば、すぐにOUT1ラインメモリに渡される。OUT1ラインメモリのデータは、OUTラインメモリが空いていれば、すぐにOUTラインメモリに渡り、このデータが外部に出力される。

【0111】このように、CONVW、OUT2、OUT1、OUTの各ラインメモリはライン単位のFIFOバッファとして動作する。

【0112】(図22に沿った説明) 図22は復号化プログラムのフローの一例を示す。このフローに沿って、復号化コマンド処理を説明する。

【0113】処理3001では初期設定として、復号化コマンド処理に必要なパラメータをRAM200内のパラメータレジスタセット202 (CH0) または204 (CH1) から、ワーキングレジスタ500にロードする。

【0114】処理3002で1ラインの復号化を行なわせるが、この処理については後に図23を参照し詳細に説明する。

【0115】処理3003でDECODEアドレスレジスタのAフラグをチェックし、1ラインの復元が終了したかどうかを判断する。処理3004でCONVRラインメモリが空いているかどうかを判断する。DECODEアドレスレジスタのA=1かつCONVRアドレスレジスタのA=0であれば、処理3005でアドレスレジスタCONVRとアドレスレジスタD0RまたはD1Rの内容を交換し、続いて処理3006でアドレスレジスタD0RまたはD1RとアドレスレジスタDECODEの内容を交換してラインメモリ間のデータの受け渡しを行なう。

【0116】これにより、ラインメモリD0RまたはD1Rに今復元したデータが渡り、次のラインを復元する準備ができる。CONVRラインメモリに参照ラインとして使い終わったデータが渡り、DECODEラインメモリに変換を終了したデータのメモリ領域が渡される。これで次のラインを復元するための準備と次の画像変換をする準備ができる。

【0117】処理3007、3008で、変換すべきデータの有無とCONVWラインメモリの空さをチェックする。

【0118】処理3009で、画像変換部1400によって主走査方向の画像変換を行なわせる。この処理の内容は図19の処理2008と同じである。また、副走査方向変換が必要な場合は、主走査変換に続いて、間引きラインの判定とフラグ制御が行なわれるが、その内容については後に詳述する。

【0119】処理3010はラインバッファの制御である。その詳細は、図24を参照して後述する。

【0120】処理3011で出力すべきデータの有無をチェックする。処理3012でOUTラインメモリのデータの外部出力を行なう。処理3013、3014では、イメージバス側のDMA転送の要求があるときに、その転送処理を行なう。これは図19に示された処理2002、2003と同じである。

【0121】処理3015で復号化コマンドの終了判断を行ない、終了条件を満足していなければ処理3002に戻る。終了条件を満足したときは、処理3016で、次の復号化コマンドに備えてCONVRラインメモリに移ったデータをラインメモリD0RまたはD1Rに戻す。

【0122】(図23に沿った説明：1ライン符号化) 図23は図22の1ライン復号化処理3002のフローである。処理3101で復号器1300の動作中を示す状態信号をチェックする。動作中でないときは、起動されたか否かを判断するため、処理3102で復号器1300の起動フラグをチェックする。起動フラグがセットされている(起動済み)であることを確認した場合は処理3103に進み、起動フラグがリセットされている場合は処理3115に進む。また復号器1300が動作中である時は直ちにリターンする。

【0123】復号器1300が動作中でない場合の処理フローは次のとおりである。処理3115で、復号器1300の起動の準備のために、内部バスDMA制御部300内のアドレスカウンタA、アドレスカウンタIまたはJに、ワーキングレジスタ500上のアドレスレジスタDECODE、アドレスレジスタD0RまたはD1Rの内容をそれぞれロードする。

【0124】以後、復号器1300からのDMA転送要求に応じて1ワードアクセスする毎に、これらのアドレスカウンタは自動的にインクリメントされ、復元データの書き込みアドレスと参照ラインデータの読み出しアドレスを指定する。

【0125】処理3116で、復号器1300のレジスタ1324と参照ライン変化画素検出部600の内部レジスタ(図示されていない)に、1ラインのワード数を設定する。このような準備の後、処理3117で復号器1300を起動し、処理3118で復号器1300の起

動フラグを”1”にセットしリターンする。以上がラインの先頭処理である。

【0126】処理3103以降は、復号器1300が1ライン分のデータを復号し、復元データがDECODEラインメモリに得られた後の処理部分である。

【0127】処理3103では、処理3118でセットした起動フラグをリセットする。処理3104で復号エラーがあったかどうかを示す復号器1300の状態信号をチェックする。

【0128】復号エラーがあれば処理3119で、復号エラー処理を行なう。例えば、エラーがあったラインを直前のラインまたは白ラインで置換する処理を行なう。復号エラーがなければ、DECODEラインメモリにエラーのないデータが復元されているので、それを示すために処理3105でDECODEアドレスレジスタのAフラグを”1”にセットする。

【0129】処理3106では、復元したラインが白ライン（全画素が白）であることを示す復号器1300の状態信号をチェックする。復号器1300は、1ワード復元する毎に比較器1320（図7）で白データかどうかを確認し、1ライン復号終了した時点で、そのラインが白ラインであったことを状態信号からマイクロプログラム側で確認できる。白ラインであるときは、処理3107で、1ページの上端の連続した白ラインを計数するためのカウンタ、またはページの下端の連続した白ラインを計数するためのカウンタ（いずれのカウンタもワーキングレジスタ500上に用意される）をインクリメントする。

【0130】処理3108ではRTC符号を検出したかどうかを示す復号器1300の状態信号をチェックする。処理3109で、復元したデータを外部に出力するかどうかを判断する。この判断は、システムバス制御部400内のレジスタセット402（CH0）または404（CH1）中の特定レジスタを参照することによって行なう。このレジスタのビット内容は、プロセッサブロック13のMPUによって設定されている。

【0131】出力しないラインであるときは、処理3110で、DECODEアドレスレジスタのBフラグを”1”にセットする。Bフラグがセットされたラインは、データ出力時に、データが無視され外部に出力されない。このような制御によって、MPU側でページの上端または下端の白ラインをカットさせるような制御を行なうことができる。

【0132】処理3112で、MPUによって設定された連続処理するライン数をデクリメントし残りライン数を求める。そして処理3113で、残りライン数をチェックし、これが0であれば、処理3114でDECODEアドレスレジスタの最終ラインを示すCフラグを”1”にセットする。残りライン数が0でないときは、直ちにリターンする。

【0133】処理3120では、復元ライン数のカウンタ（ワーキングレジスタ500上に用意される）をインクリメントする。このカウンタ値で1ページのライン数が得られる。このライン数は、1ページの処理が終わった段階で、マイクロプログラムによってRAM200の対応チャンネル用の復号化コマンドパラメータレジスタ領域210に保存される。この領域はMPUから直接アクセスすることができる。

【0134】（図24に沿った説明：ラインバッファ制御）図24は図22の処理3010（ラインバッファ制御）のフローである。処理3201、3202で、CONVWアドレスレジスタのA=1かつOUT2アドレスレジスタのA=0であることを確認し、処理3203でアドレスレジスタCONVW、OUT2の内容入れ替えを行なう。

【0135】これで、CONVWラインメモリのデータがOUT2ラインメモリに入り、CONVWラインメモリには空き領域が渡る。アドレスレジスタのフラグの状態は、OUT2はA=1、CONVWはA=0となる。

【0136】処理3208、3204、3205でラインメモリOUT2、OUT1の間のデータ交換を行ない、処理3209、3206、3207でラインメモリOUT1、OUTの間のデータ交換を行なう。処理3207でOUTアドレスレジスタのAフラグは”1”にセットされる。以上の処理でCONVW、OUT2、OUT1、OUTの各ラインメモリがライン単位のFIFOバッファとして利用されることになる。

【0137】（図25に沿った説明：画像データ出力）図25は、図22の画像データ出力処理3012のフローである。処理3301でイメージバス制御部3301が動作中であるか否かチェックし、動作中であればリターンする。動作中でない場合、処理3302でイメージバス制御部100の起動フラグがセットされているか調べる。起動フラグが”1”であればイメージバス制御部100は起動後であって、しかも動作中でないということであるので、処理3311以下のライン端処理に進む。

【0138】起動していなければライン先頭処理に入る。処理2303でOUTアドレスレジスタのBフラグをチェックし、OUTラインメモリのデータを出力するかどうかを判断する。

【0139】Bフラグが”1”でなければ出力すべきデータであるので、そのDMA転送による出力の準備をする。まず処理3304で、OUTアドレスレジスタより、内部バスDMA制御部300内のOUTラインメモリ対応のアドレスカウンタ302に先頭アドレスをロードする。処理3305で、イメージバス制御部100内のアドレスレジスタ104の1つに外部RAM28のアドレスを設定する。処理3306で、イメージバス制御部100内の転送ワード数レジスタ114の1つに出力

ラインのワード数を設定する。処理3307で、イメージバス制御部100の動作モードを設定する。ここではメモリライト・モードに設定することになる。そして処理3308でイメージバス制御部100を起動する。処理3309でイメージバス制御部100のメモリライト・モードの起動フラグを”1”にセットし、リターンする。

【0140】処理3303でBフラグが”1”であった場合、データ出力は行わず、処理3310でOUTアドレスレジスタのAフラグを”0”にリセットすることで、OUTラインメモリの内容を無視させる。この処理でラインの間引きが達成される。

【0141】処理3311以降はライン端の処理である。処理3311では、処理3309でセットしたイメージバス制御部100の起動フラグをリセットする。処理3312で、OUTアドレスレジスタのEフラグをチェックすることにより、出力ラインが副走査方向の拡大対象ラインかどうかを判断している。

【0142】拡大対象でなければ(E=0)、OUTラインメモリのデータの2回目の出力は不要であるので、処理3313でOUTアドレスレジスタのAフラグを”0”にリセットし、OUTラインメモリを解放する。拡大対象ラインであれば(E=1)、処理2314でOUTアドレスレジスタのEフラグをリセットする。そのAフラグは”1”であるから、このデータは再度出力され、結果として副走査方向の拡大(ラインの補間)が達成される。処理3315で、次のラインを出力するために、外部RAM28のアドレスを更新しリターンする。

【0143】復号化コマンドの終了判定は、Cフラグが”1”にセットされたデータを出力したかどうかで判断する。Cフラグ(最終ラインフラグ)はアドレスレジスタの交換によって、次のように伝搬する。

DECODE, C=1→CONVR, C=1

CONVR, C=1ならば画像変換終了後 CONW, C=1→OUT, C=1

(伸長動作のまとめ)以上に説明したように、内部RAM200上のラインメモリにデータが詰まった後は、復号化処理(処理3002)、画像変換処理(処理3009)、画像データ出力処理(処理3012)が並列に動作する。更にイメージバス側のDMA転送処理(処理3013、処理3014)も、これらと並列動作ができる。

【0144】したがって、本圧縮伸長装置1の伸長処理時間は近似的に次式で表わすことができる。

処理時間= max {復号化時間, 画像変換時間, 画像データ出力時間}

図26は復号化コマンド処理でのラインメモリの使い方を示している。この図からわかるように、DECODE, D0R/D1R, CONVRの各ラインメモリは循環的に使用され、CONVW, OUT2, OUT1, O

UTの各ラインメモリも循環使用される。

【0145】符号変換動作の説明

次に、符号変換動作について説明する。この符号変換とは、ある符号データを入力して、それを別の符号データに変換することである。例えば、MR符号からMMR符号への変換である。

【0146】符号変換動作の場合、変換すべき符号データはシステムバス10から入力し、復号器1300によって復号され、復元データがDECODEラインメモリに書き込まれる。既に説明した復号化動作によって、CONVRラインメモリのデータが画像変換される。ここまでは伸長動作と全く同一である。

【0147】その後、CONVWラインメモリのデータが符号化の対象となる。これ以後は圧縮動作と全く同じである。

【0148】以上の復号化、画像変換、符号化の処理を順に1ライン毎に繰り返して1ページ分の符号データを別の符号に変換することができる。

【0149】この符号変換処理の場合、既に説明したように、図15に示すアドレスレジスタ502とラインメモリ216が定義される。図28はラインメモリの使い方の説明図である。

【0150】図27は符号変換プログラムのフローである。図19または図22のフローと同じ番号の処理は同内容であるので、その説明を省略する。

【0151】図27のフローから、次のことが容易に理解される。

a) 復号化処理、画像変換処理、符号化処理、イメージバス側の2つのDMA転送処理が並行動作する。

b) 復号エラーが生じても、1ライン復号化処理(処理3002)内で復号エラーのチェックと復号エラー処理(図23の処理3119)が行なわれるので、エラー処理後の画像データに対して符号化が行なわれる。したがって、変換後の符号データは復号エラーを含まない。

【0152】なお、符号変換動作または伸長動作において、復号化処理(処理3002)で1ページのライン数、ページの上端と下端の連続した白ライン数が求められ(図23の処理3120, 3107)、動作終了時点でRAM200上の復号化コマンド用パラメータレジスタ領域210に保存される。MPU13は、これらのパラメータを読み出し、拡大縮小率やページ上下端のカットラインの決定などのために用いることができる。

【0153】副走査方向の画像変換(縮小)の説明

副走査方向の縮小は一定ライン数毎に1ラインの間引くことで実現され、副走査方向の拡大は一定ライン数毎に1ラインをコピー(補間)することで実現される。ここでは、副走査方向の縮小動作について、間引きラインの決定方法を中心に詳細に説明する。

【0154】図29は副走査方向変換動作の処理の概念図である。関連したパラメータ(ワーキングレジスタ5

00上に用意されるレジスタ551～557に置かれる)は次の通りである(ただし、チャンネル0用)。

C0-VCONV(副走査変換率):レジスタ551

C0-ZLINE(一定ライン数、後に説明する):レジスタ552

C0-VCWRK(ワークレジスタ):レジスタ553

C0-ZLWRK(ワークレジスタ):レジスタ554

レジスタX(RAM200のアドレスを指すレジスタ):レジスタ555

非白ワードカウンタ:レジスタ556

N(定数):レジスタ557

(白ラインを優先しないアルゴリズムの説明) CONVWラインメモリに1ラインの有効データが得られる毎に、16ビットALU1202でC0-VCONVの積算を行なう。この積算値がC0-VCWRKである。積算した時にALU1202がオーバフローしてキャリーが出たラインが間引きの対象になる。

【0155】16ビットのALU1202のキャリーは、 $65536 / (C0-VCONV)$ ライン毎に1回出る。 $M = 65536 / (C0-VCONV)$ とすると、Mラインに1ライン間引かれるので、縮小率Rは $R = (M-1) / M = 1 - (1/M)$ となる。

【0156】この関係から、MPU13は、縮小率RよりC0-VCONVを決定し圧縮伸長装置1に対して設定する。

【0157】図30(a)はC0-VCONVの積算過程の説明図で、画像のラインと各ラインでの積算値C0-VCWRKを左右に並べて示している。ここに示した例では、第5ラインでキャリーが出るので、このラインが間引きラインとされる。

【0158】このようなアルゴリズムは従来から知られているものである。なお、副走査方向の拡大の場合のコピーラインの決定も同様のアルゴリズムによって行なうことができ、その積算過程を図30(b)に示す。この例では、第5ラインでキャリーが出るので、第5ラインが拡大対象ラインとなり、これが2回符号化されることになる。

【0159】本圧縮伸長装置1においては、上記アルゴリズムと以下に説明する改良されたアルゴリズムのいずれも選択できる。

【0160】(改良されたアルゴリズムの説明) 図31は、この改良された副走査方向縮小アルゴリズムの説明図である。矢印はキャリーの出るラインの位置を示している。上述の従来のアルゴリズムによれば、キャリーが出たラインが無条件に間引かれるため、細い野線が失われ画質が劣化する欠点があった。

【0161】この改良アルゴリズムは、白ラインを優先して間引くことで、そのような画質劣化を防ぐ。すなわち、C0-ZLINEをライン間引き間隔よりも小さく

設定し、キャリーが出たラインからC0-ZLINEの範囲で白ラインを捜し、それを間引く。C0-ZLINEの範囲に白ラインがなければ、この範囲の最後の(C0-ZLINE目の)ラインを間引く。

【0162】ここで言う白ラインとは、非白ワード数 $\leq N$ を満たすラインのことである。NはMPUが設定する値である。N=0の場合、全ワードが白ワードのラインを白ラインとして捜すことになる。

【0163】(図32、図33に沿った説明) 図32は間引きラインを判定しフラグを設定する処理のフローである。この処理は、マイクロROM1601に書かれたマイクロプログラムによって実現されるもので、CONVWラインメモリに1ラインのデータが生成される毎に(図22または図27の処理3009あるいは図19の処理2008で、主走査変換処理が終了した直後に)実行される。ここで設定されたフラグの情報に従って、後続の復元データ出力処理(復号化動作の場合は図22の処理3012)で実際にラインの間引きが行なわれる。

【0164】図32において、処理4001でC0-ZLINEをデクリメントする。処理4002、4003でC0-VCONVとC0-VCWRKに加える(積算値の更新)。処理4004でALU1202のキャリー(ALU1202の状態信号に含まれる)が出たか否かをチェックする。

【0165】キャリーが出ているときは処理4005Aで、キャリーが出ていないときは処理4005Bで、MPUによるシステムバス制御部400内の特定レジスタ(402、404)への設定が、白ライン優先間引き(上述の改良アルゴリズム)であるかチェックする。処理4005Bで白ライン優先間引きでないか判断したとき(上述の従来アルゴリズムが選択されているとき)はリターンし、白ライン優先間引きであると判断したときは処理4007へ進む。

【0166】処理4005Aで白ライン優先間引きであると判断した場合、処理4006で、C0-ZLWRKに初期値C0-ZLINEを設定し、処理4007に進む。処理4005で白ライン優先間引きであると判断したときは、直ちに処理4007に進む。

【0167】この処理4007はCONVWラインメモリのデータをチェックし、間引きラインとしての条件を満たしているかどうかを判断するもので、その内容は図33に示されている。この判断結果を次の処理4008でチェックする。白ラインと判断された場合は、処理4010でC0-ZLINEの範囲を調べ終わったかどうかを判断し、調べ終わっていないときはリターンする。

【0168】C0-ZLINEの範囲内を調べ終わつたと判断された場合、または処理4008で白ラインと判断された場合には、処理4009でCONVWアドレスレジスタのBフラグを"1"にしリターンする。これで、後続処理(図22の処理3013)で、CONVW

ラインメモリのデータを間引き（縮小対象）ラインとして扱われることになる。

【0169】図33は処理4007のフローである。処理4101でCONVWラインメモリの先頭アドレスを、レジスタXに設定する。処理4102で、ALU1202のアキュムレータ（ACCA）に、レジスタXが指すアドレスの内容をロードする。これは、内部データバス1800（図5）を通してRAM200からデータを1ワード（16ビット）読み出す処理である。

【0170】処理4103で、そのデータが16ビット全部が白画素の白データであるか、1ビットでも黒画素を含む非白データであるかを判断する。非白データのときは処理4104で非白ワードカウンタ（レジスタ556）をインクリメントする。処理4106で1ラインのチェックが終了したかどうかを判断し、未終了のときは処理4105でレジスタXをインクリメントし処理4102から処理を繰り返す。

【0171】1ラインのチェックが終了すると、処理4107で、非白ワードカウンタの計数値が設定値N以下かどうかを判定する。N以下であれば、処理4108で白ラインであることを示すフラグをセットする。処理4008（図32）では、このフラグをチェックする。

【0172】なお、ここまでの説明から明かなように、本圧縮伸長装置1は、画像変換だけを行なわせることもできる。また、データを全く出力させずに伸長動作を行なわせ、符号データのエラーチェック、あるいはライン数やページ上下端白ライン数の検出を高速に行なうことも可能である。

【0173】本圧縮伸長装置1の処理時間の実測値を図34と図35に示す。図34は圧縮動作（符号化処理＋画像変換処理）の1ラインの処理時間を示し、図35は伸長動作（復号化処理＋画像変換処理）の1ラインの処理時間を示している。

【0174】測定条件は、画像変換処理の変換率を100%として圧縮伸長装置内部の負荷が最大になる状態に設定し、また1ラインの長さをB4サイズ相当の2048画素とし、マシンサイクルを100nsec（10MHz）としている。測定値は、ファクシミリ標準符号化方式のMH、MR、MMR、符号化データ量が画像データ量を超えない前記MG3符号化方式、それに参照ラインが全白、符号化ラインが1ビット交番の画像データH（W1，B1）について示されている。

【0175】横軸の数字は1ワード（16ビット）内の変化画素数を意味する。平均的な文書画像の場合、1ワード内の変化画素数は0～4程度の範囲である。したがって、平均的文書画像に対しては、1ラインを100μsec程度で圧縮または伸長することができる。

【0176】符号変換処理の処理時間は、圧縮処理時間と伸長処理時間の合計時間以下である。つまり、B4サイズからB4サイズへのサイズ変換（画像変換）の時間

を含めても、平均的文書画像の符号変換処理を200μsec／1ライン程度の時間で高速に実行できる。

【0177】実際的には、B4サイズからA4サイズへの画像変換を行なうのが普通であるので、符号変換処理時間はさらに高速である。また、ここでは10MHz（マシンサイクル100nsec）として測定したが、本圧縮伸長装置1は16MHzで動作させることも容易であることが確認されているので、符号変換処理時間は100μsec／ライン程度まで短縮でき、1ページを2400ラインとすると、0.2～0.3secの時間で1ページの符号変換が可能である。よって、前述のように1台の圧縮伸長処理装置1で16回線分の符号化処理を担うことができるわけである。

【0178】ここまで、ファクシミリメール装置の実施例について説明したが、ファクシミリ手順以外の特定の通信手順によって同様の画像符号データ送信処理を行なう画像通信装置においても、本発明を適用できる。

【0179】

【発明の効果】以上の説明から明かなように、本発明は以下の効果を有するものである。

【0180】1）高速の符号変換装置を用い、回線インターフェイスのための回線対応の通信ユニットと符号変換装置とを独立させ、1個の符号変換装置を、同時に通信手順が実行される複数回線分の画像受信端末に対する符号変換処理に共通に用いるため、高価な符号変換装置の個数を減らし、画像通信装置の構成単純化とコスト削減を達成できる。

【0181】2）N個の回線ユニットと1個の符号変換装置を組み合わせた単位ユニットの追加のみによって、N回線単位の回線増設を容易に、しかも最小のコストで行なうことができる。

【0182】3）画像符号データの符号変換処理のための複数の処理ブロックと、この複数の処理ブロックの処理実行に必要な複数のラインメモリと、外部の前記バスとのインターフェイスのためのブロックと、前記ラインメモリ及び前記各ブロックに対するデータ転送のための内部バスと、前記各ブロックを制御するブロックとを具備する改良された符号変換装置を用いることによって、1個の符号変換装置で多数回線分の符号変換処理を担わせ、画像通信装置のコストの相当大きな割合をしめる符号変換装置のコストを大幅に削減することができるため、収容回線数が多い画像通信装置も安価に提供可能となる。

【図面の簡単な説明】

【図1】本発明のファクシミリメール装置の概略ブロック図

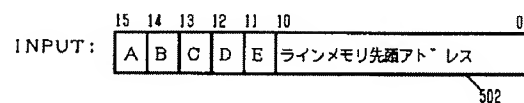
【図2】本発明のファクシミリメール装置の回線増設例を示す概略ブロック図

【図3】（a）1回線分の処理を示す概略フローチャート

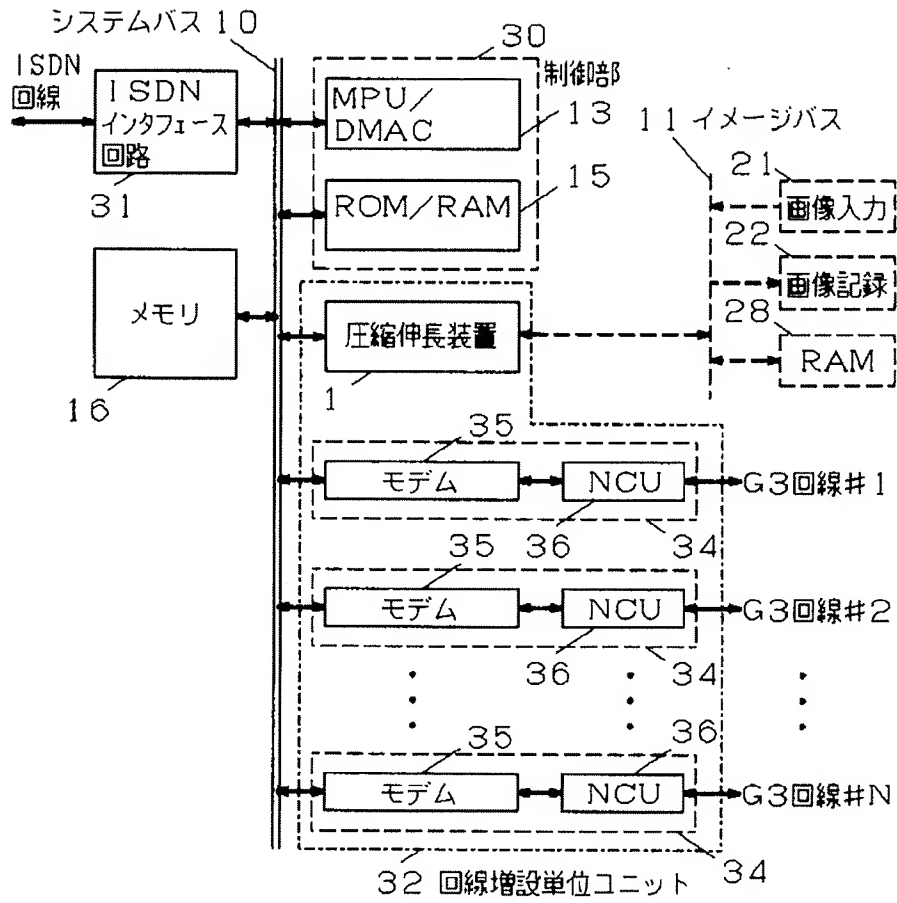
(b) 符号変換の制御を示す概略フローチャート
【図 4】本発明のファクシミリメール装置の通信シーケンス例を示す図
【図 5】圧縮伸長装置のブロック図
【図 6】圧縮伸長装置内の符号器のブロック図
【図 7】圧縮伸長装置内の復号器のブロック図
【図 8】圧縮伸長装置内の画像変換部のブロック図
【図 9】圧縮伸長装置内の算術論理演算部と周辺のプロック図
【図 10】圧縮伸長装置内のマイクロプログラム制御部とシステムバス制御部のブロック図
【図 11】圧縮伸長装置の内部RAMの使用法の説明図
【図 12】圧縮伸長装置内のイメージバス制御部のブロック図
【図 13】符号化コマンド処理のためのラインメモリ、アドレスカウンタ及びアドレスレジスタの説明図
【図 14】復号化コマンド処理のためのラインメモリ、アドレスカウンタ及びアドレスレジスタの説明図
【図 15】符号変換処理のためのラインメモリ、アドレスカウンタ及びアドレスレジスタの説明図
【図 16】アドレスレジスタの構成図
【図 17】ラインメモリ間のデータ受け渡しを実行する前の状態を示す図
【図 18】アドレスレジスタの内容交換によりラインメモリ間のデータを受け渡した後の状態を示す図
【図 19】圧縮動作のフローチャート
【図 20】画像データ入力のフローチャート
【図 21】圧縮動作時のラインメモリの使われ方とデータフローを示す図
【図 22】伸長動作のフローチャート
【図 23】1ライン復号化のフローチャート
【図 24】ラインバッファ制御のフローチャート
【図 25】イメージ出力のフローチャート
【図 26】伸長動作時のラインメモリの使われ方とデータフローを示す図
【図 27】符号変換動作のフローチャート
【図 28】符号変換動作時のラインメモリの使われ方とデータフローを示す図
【図 29】副走査方向の画像変換のためのパラメータの説明図
【図 30】(a) 副走査方向縮小のための間引きラインの決定方法の説明図

(b) 副走査方向拡大のためのコピーラインの決定方法の説明図
【図 31】改良アルゴリズムによる間引きラインの決定方法の説明図
【図 32】副走査縮小のフローチャート
【図 33】CONVWラインメモリのデータチェックのフローチャート
【図 34】圧縮伸長装置の各符号化形式と圧縮動作時の1ライン処理時間との関係を示す関係図
【図 35】圧縮伸長装置の各符号化形式と伸長動作時の1ライン処理時間との関係を示す関係図
【図 36】従来のファクシミリメール装置の概略ブロック図
【図 37】従来のファクシミリメール装置の通信シーケンス例を示す図
【符号の説明】
1 圧縮伸長装置
10 システムバス
16 メモリ
30 制御部
31 ISDNインターフェイス回路
32 回線増設単位ユニット
34 通信ユニット
35 モデム
36 NCU
100 イメージバス制御部
200 内部RAM
216 ラインメモリ
300 内部バスDMA制御部
400 システムバス制御部
500 ワーキングレジスタ
600 変化画素検出部
700 変化画素検出部
800 変化画素検出部
1200 算術論理演算部
1300 復号器
1400 画像変換部
1500 符号器
1600 マイクロプログラム制御部
1700 DMA制御バス
1800 内部データバス
1900 内部データバス

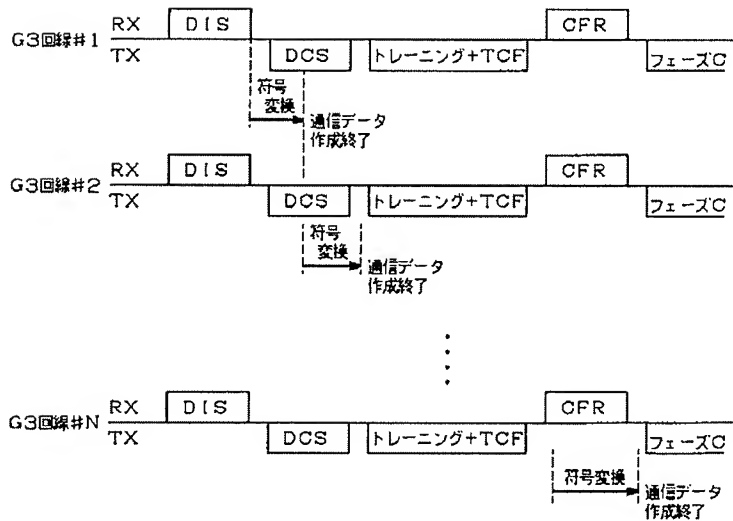
【図 16】



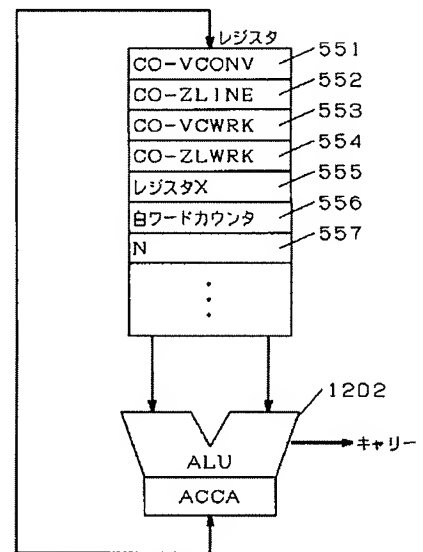
【図1】



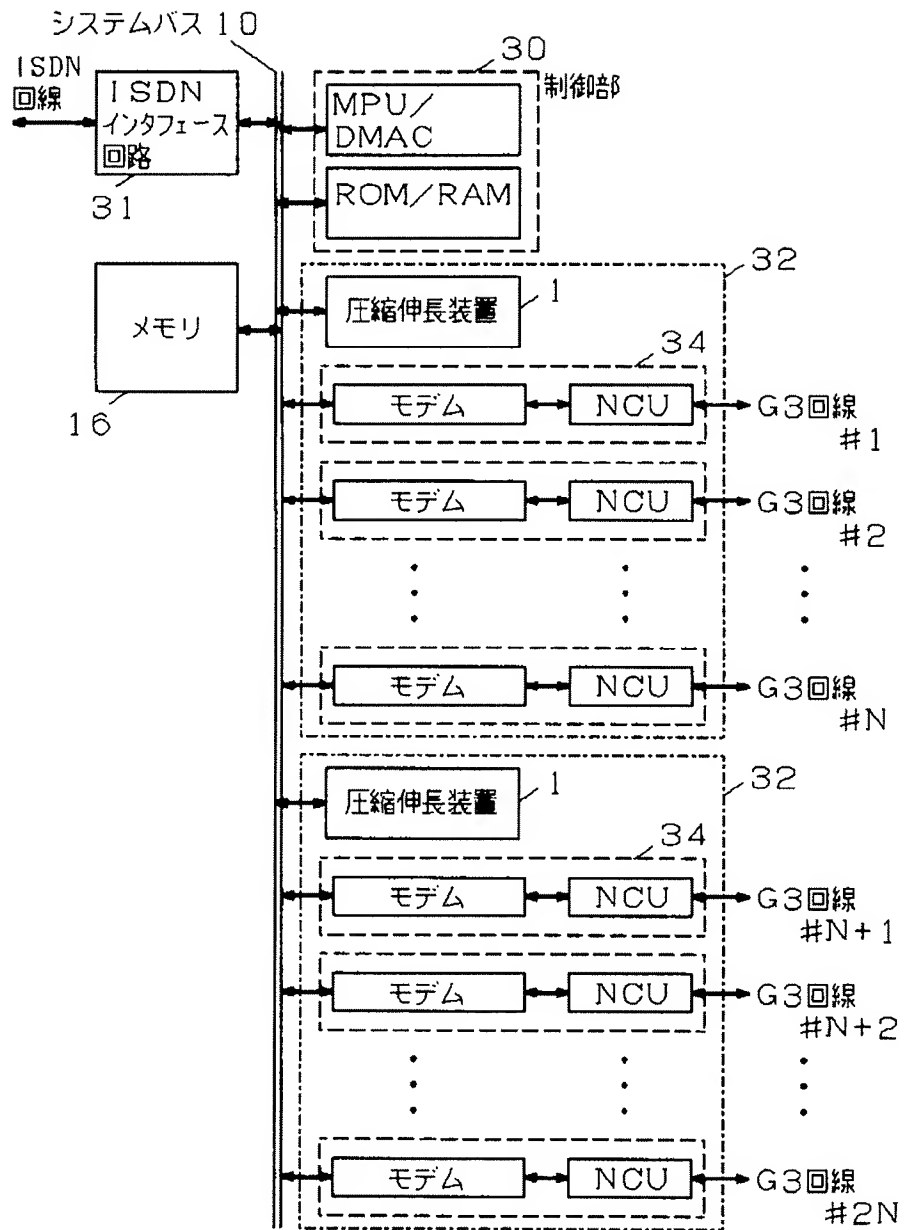
【図4】



【図29】

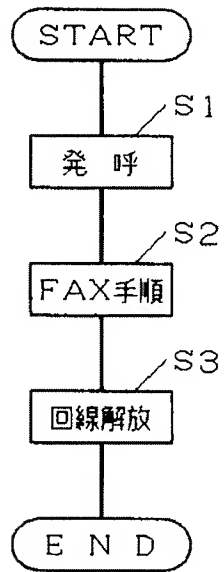


【図2】

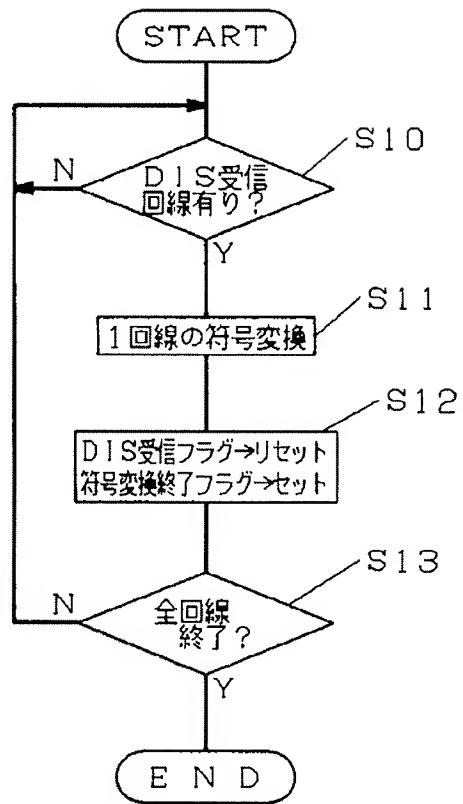


【図3】

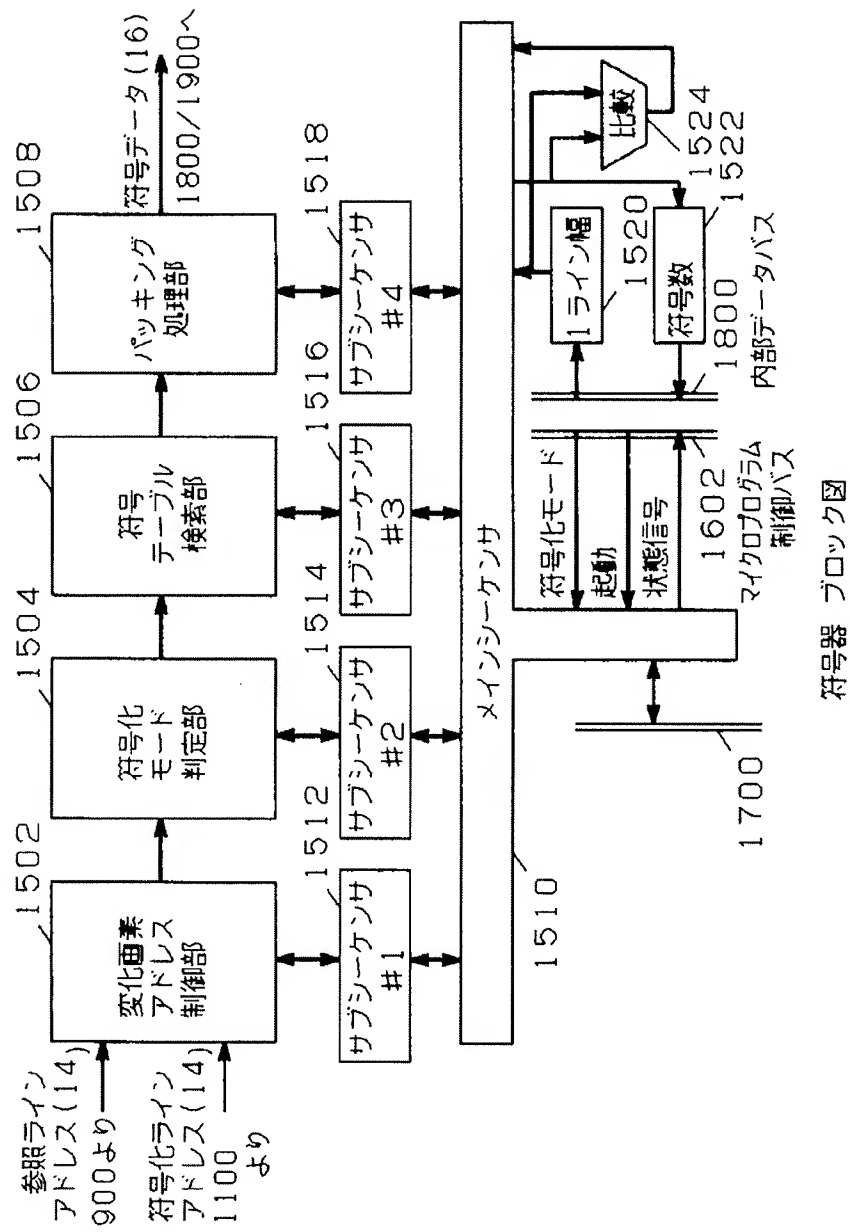
(a)

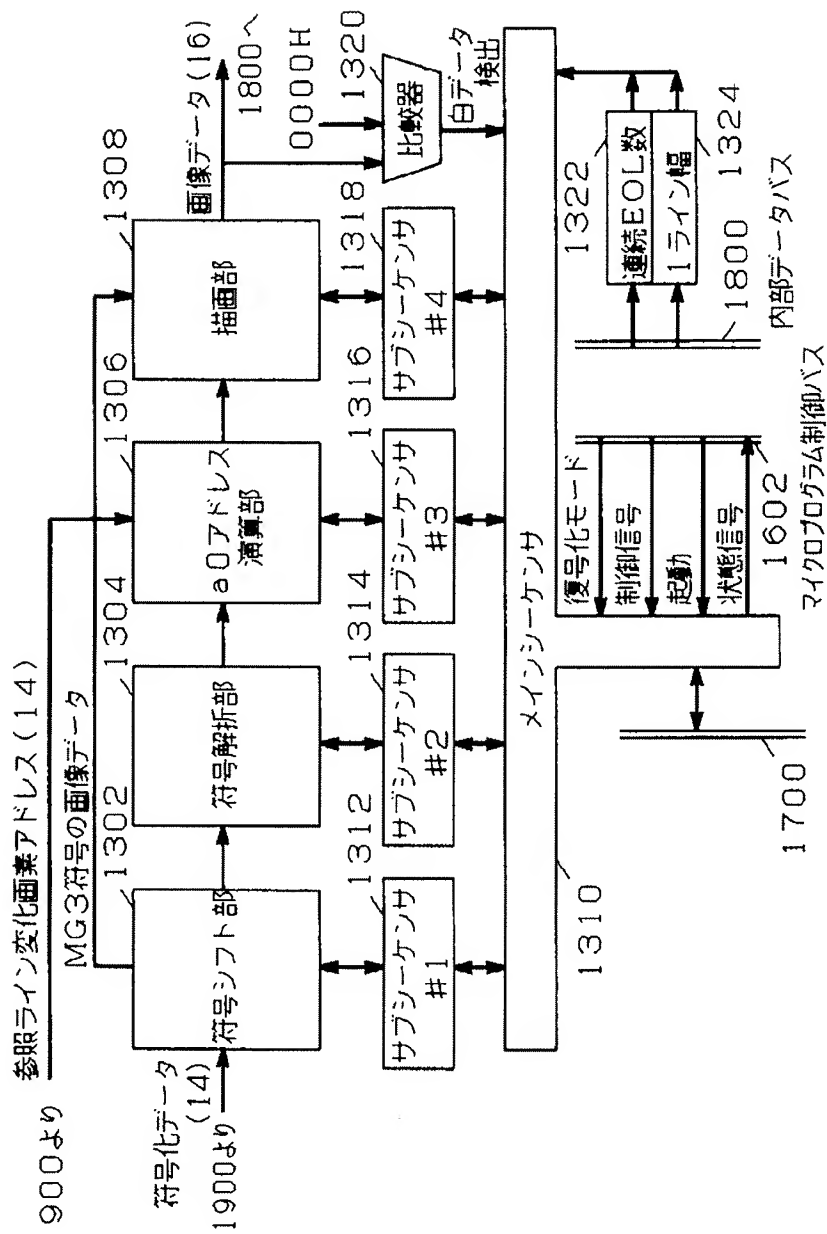


(b)



【図6】

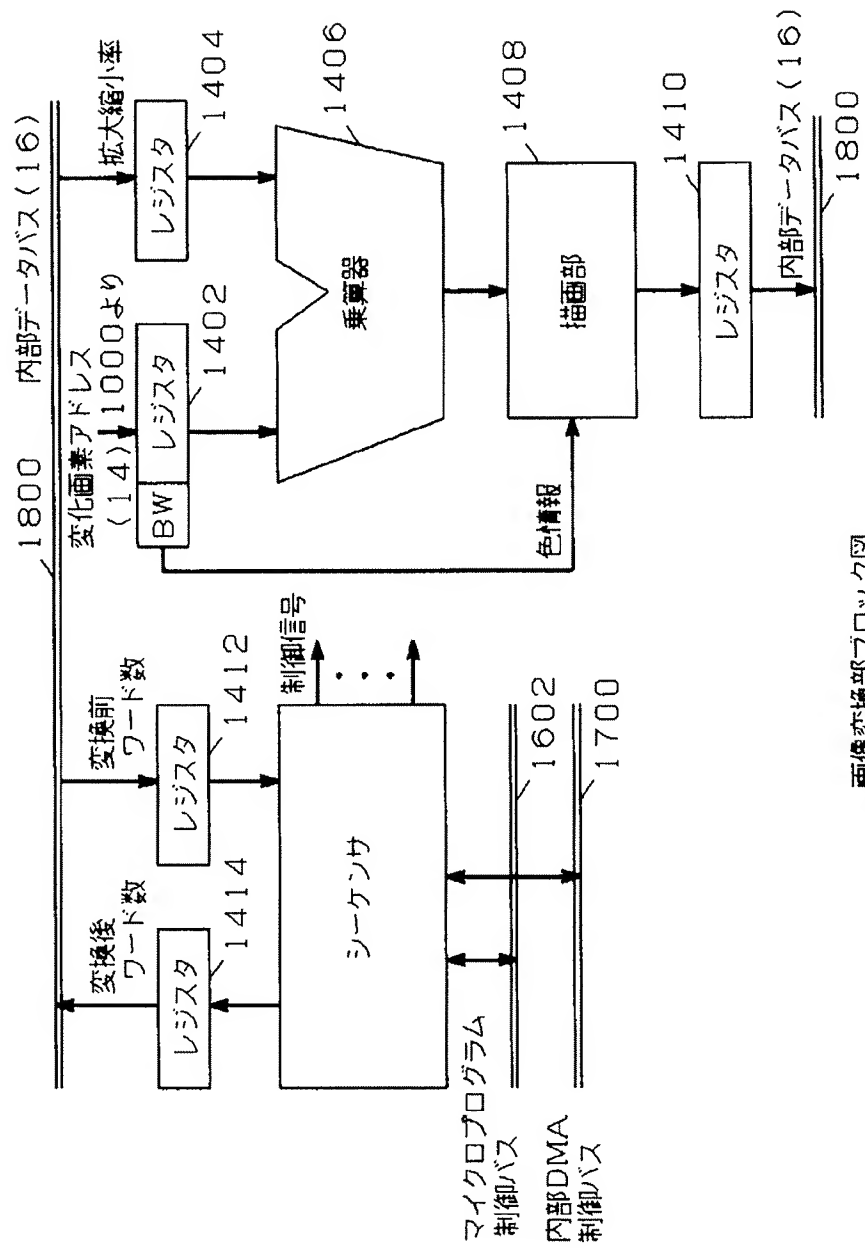




復号器 ブロック図

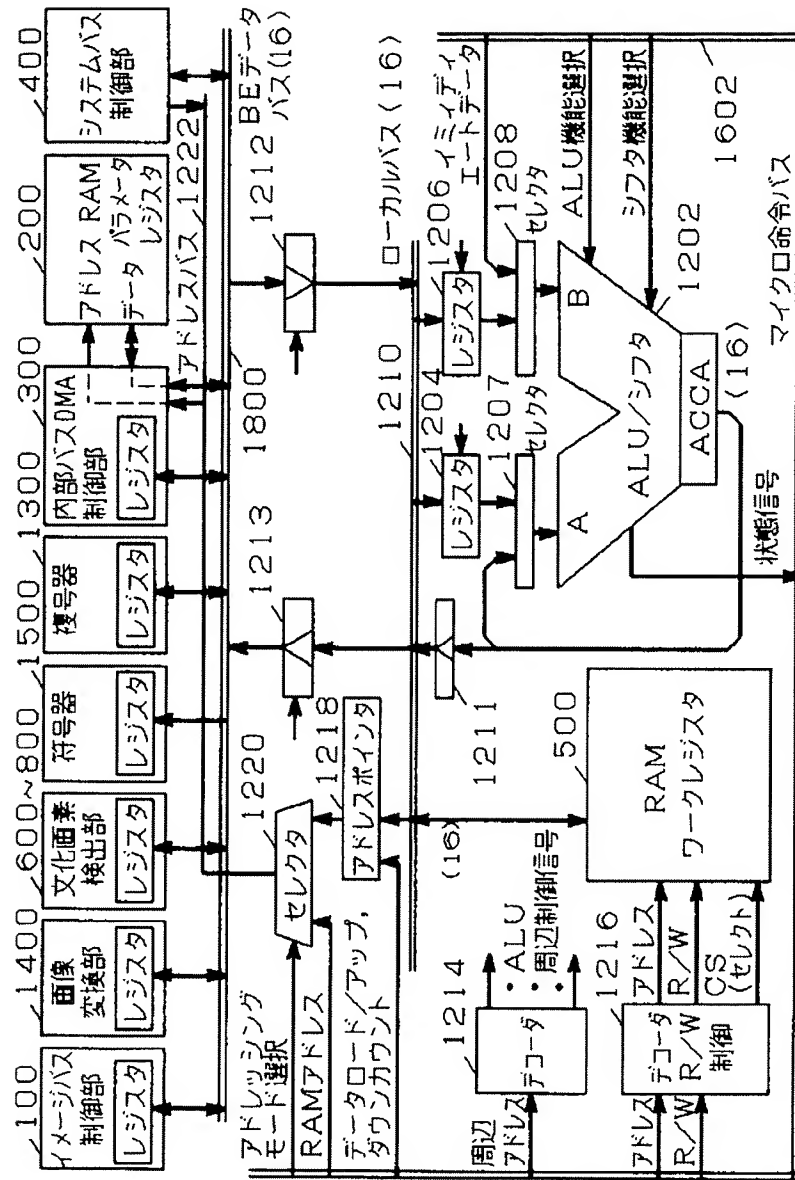
【図7】

【図8】



画像変換部ブロック図

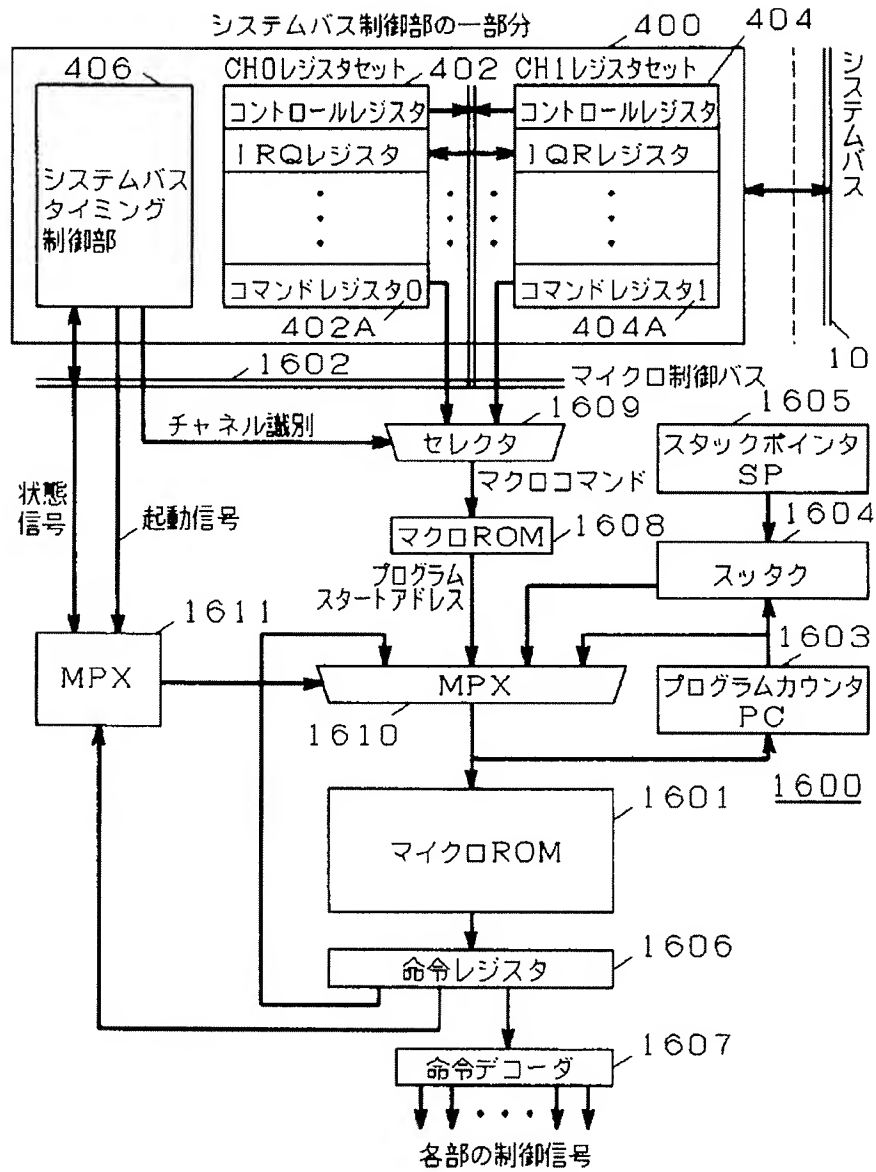
【図9】



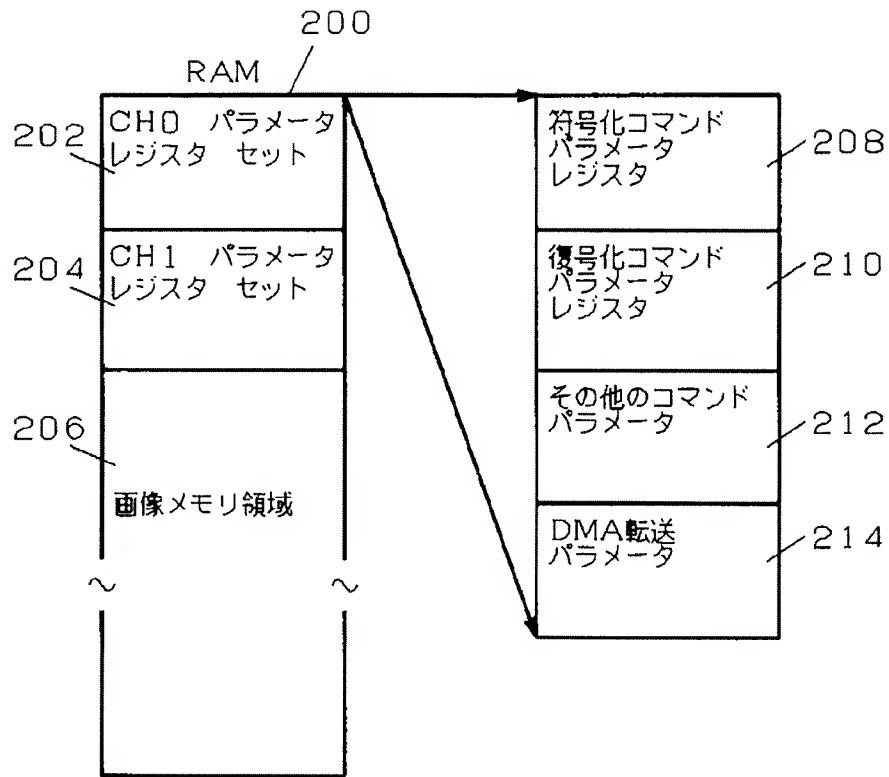
マイクロプログラム制御部

ALU周辺ブロック図

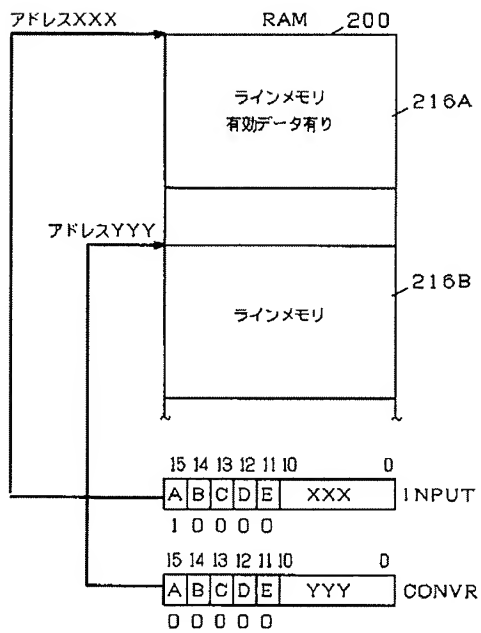
【図10】



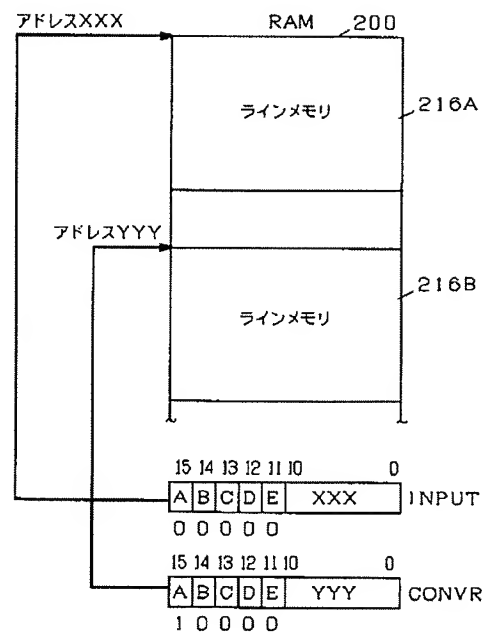
【図11】



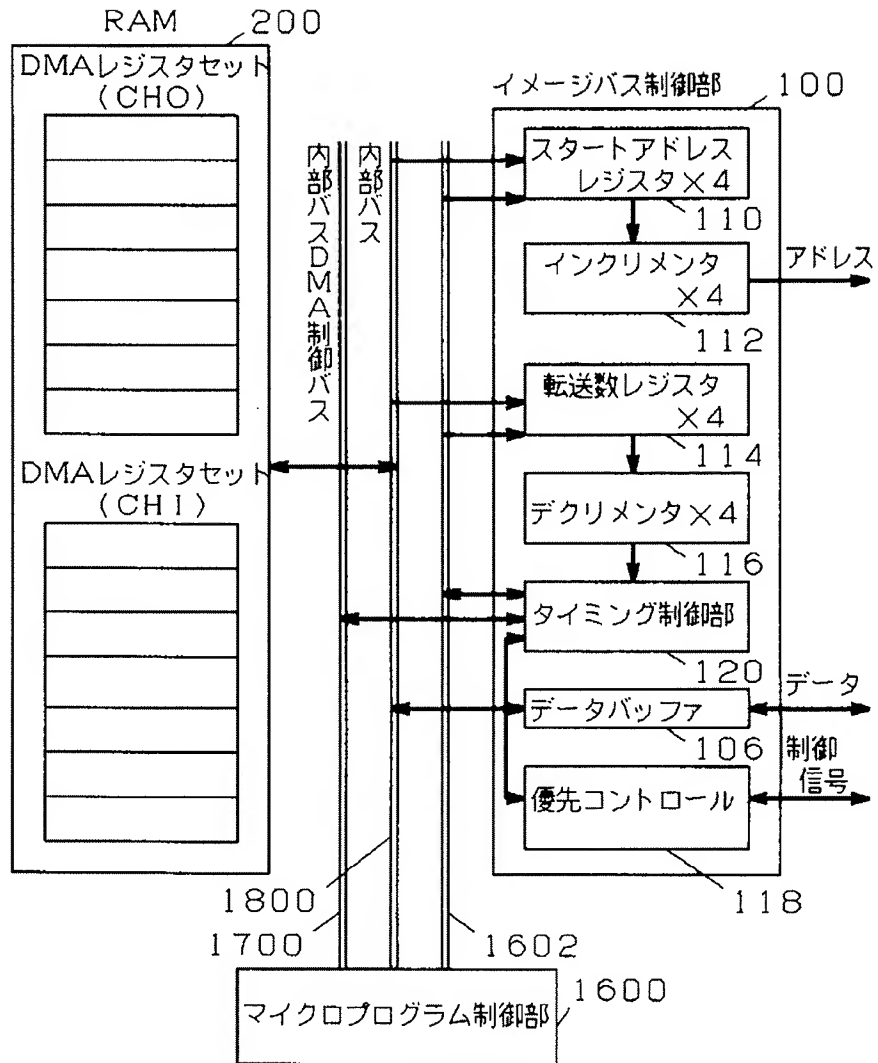
【図17】



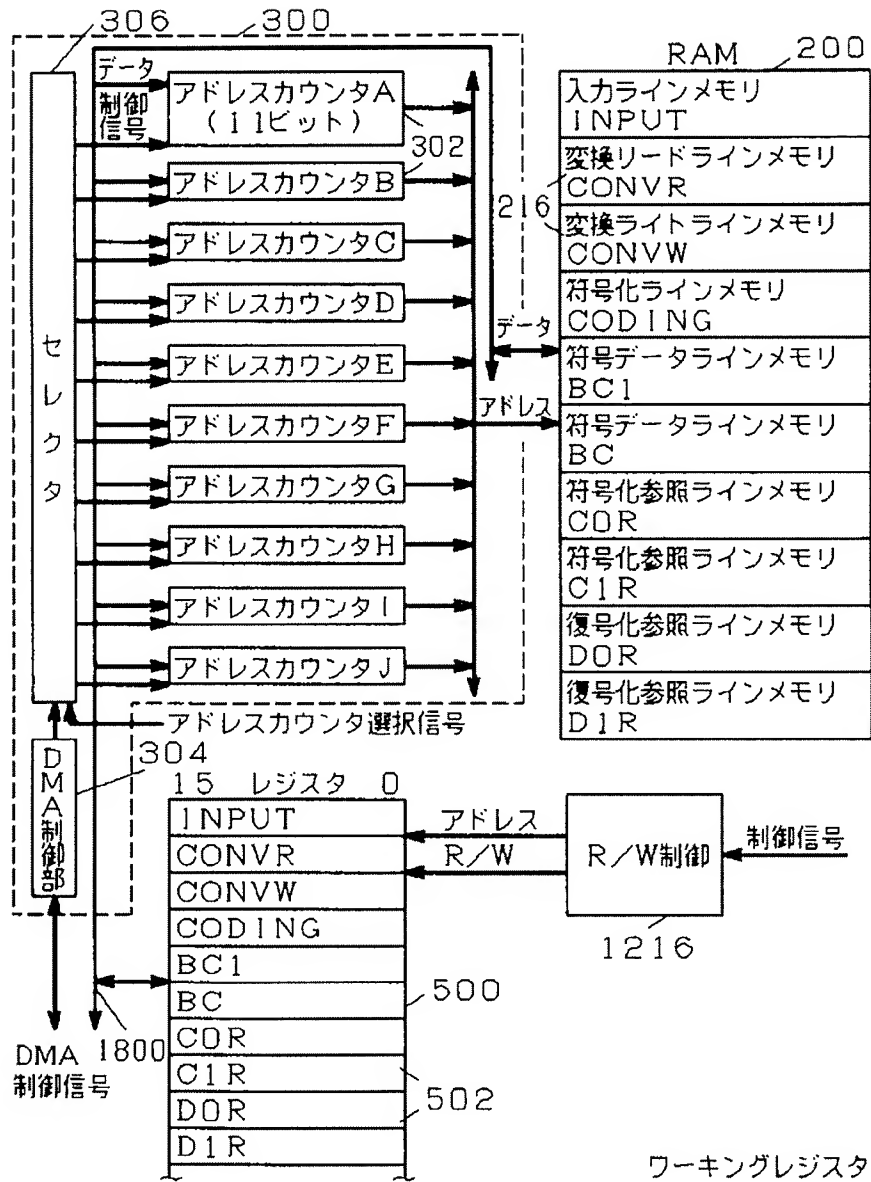
【図18】



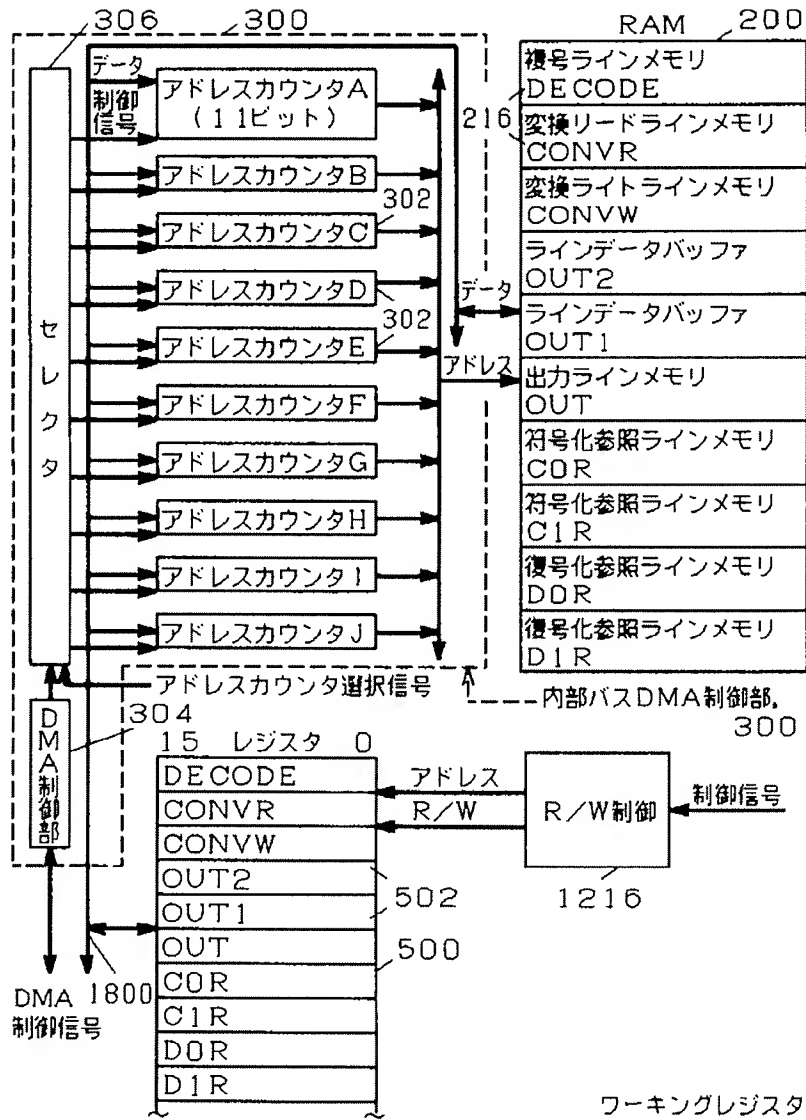
【図12】



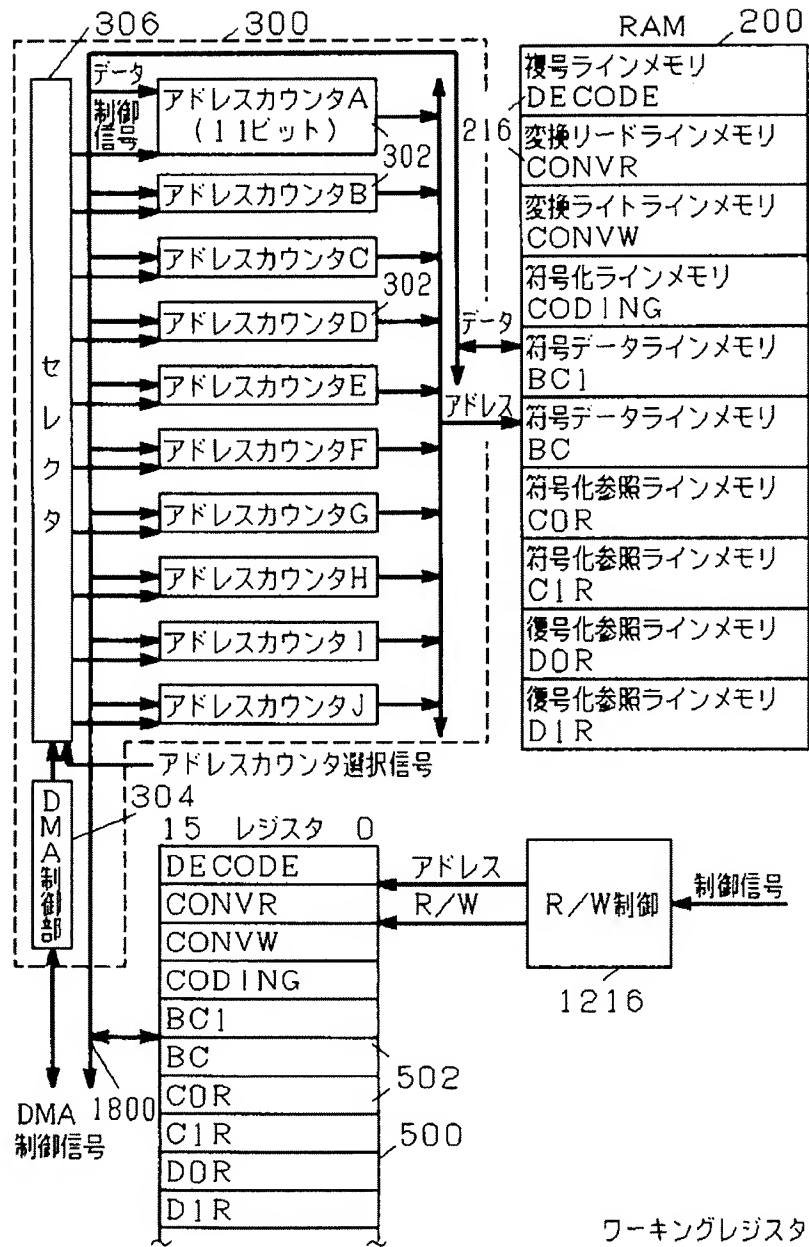
【図13】



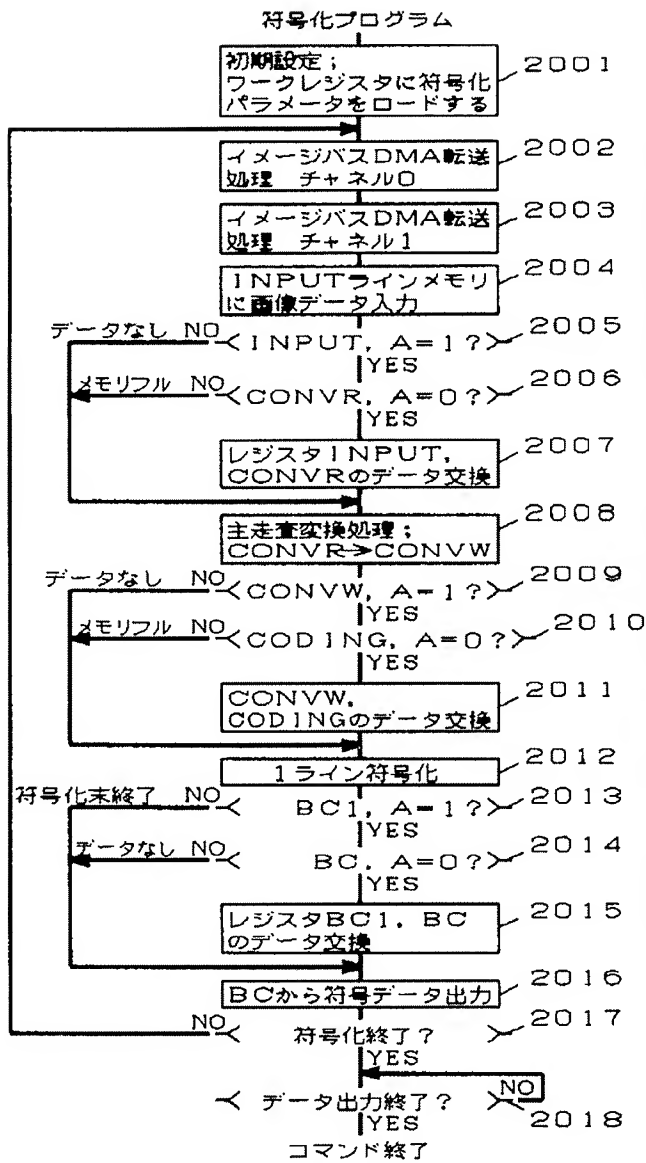
【図14】



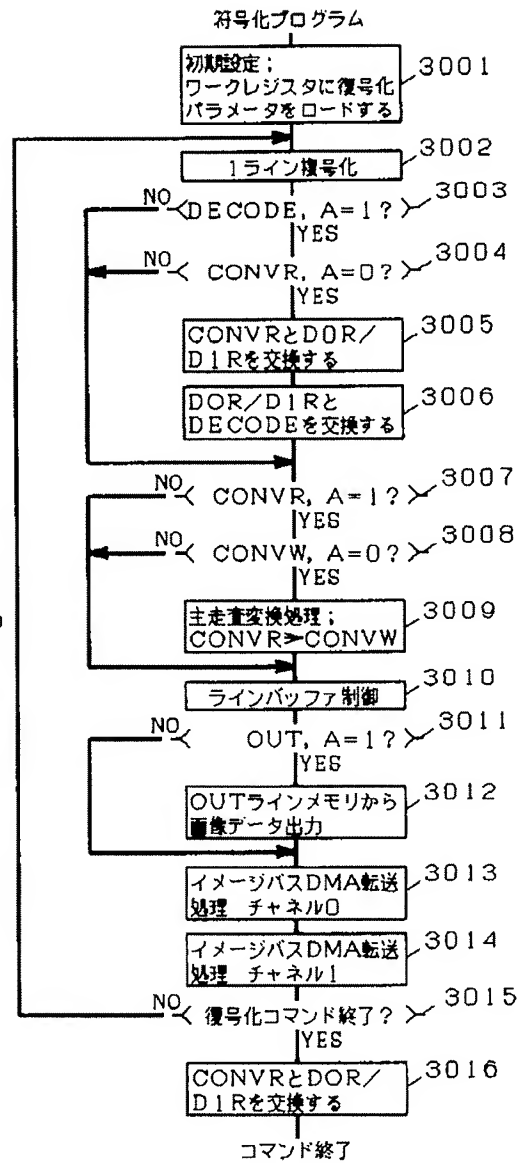
【図15】



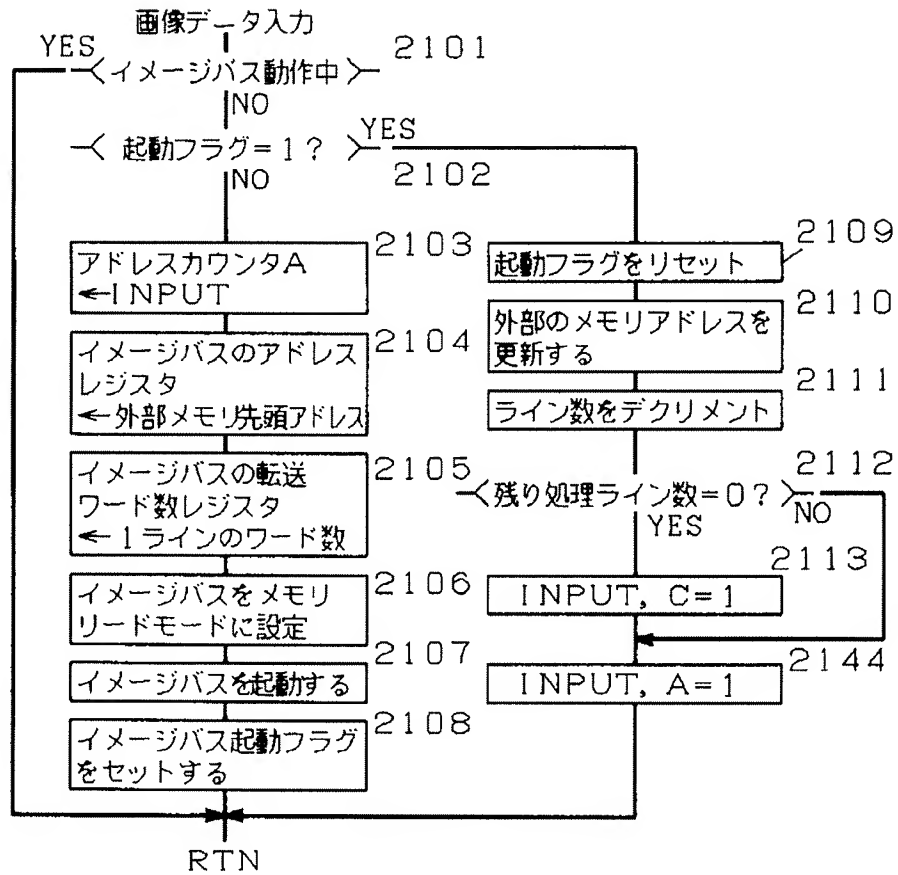
【図19】



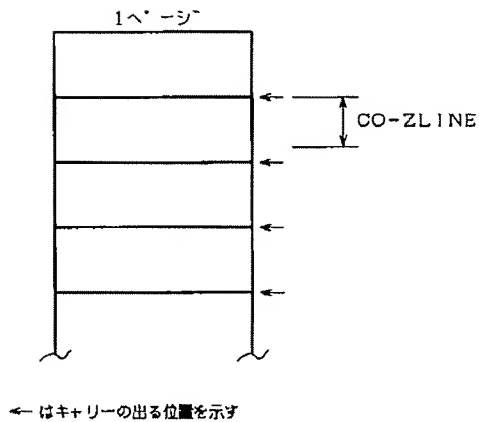
【図22】



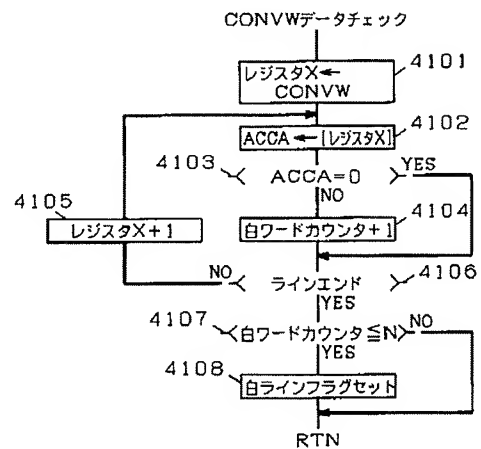
【図20】



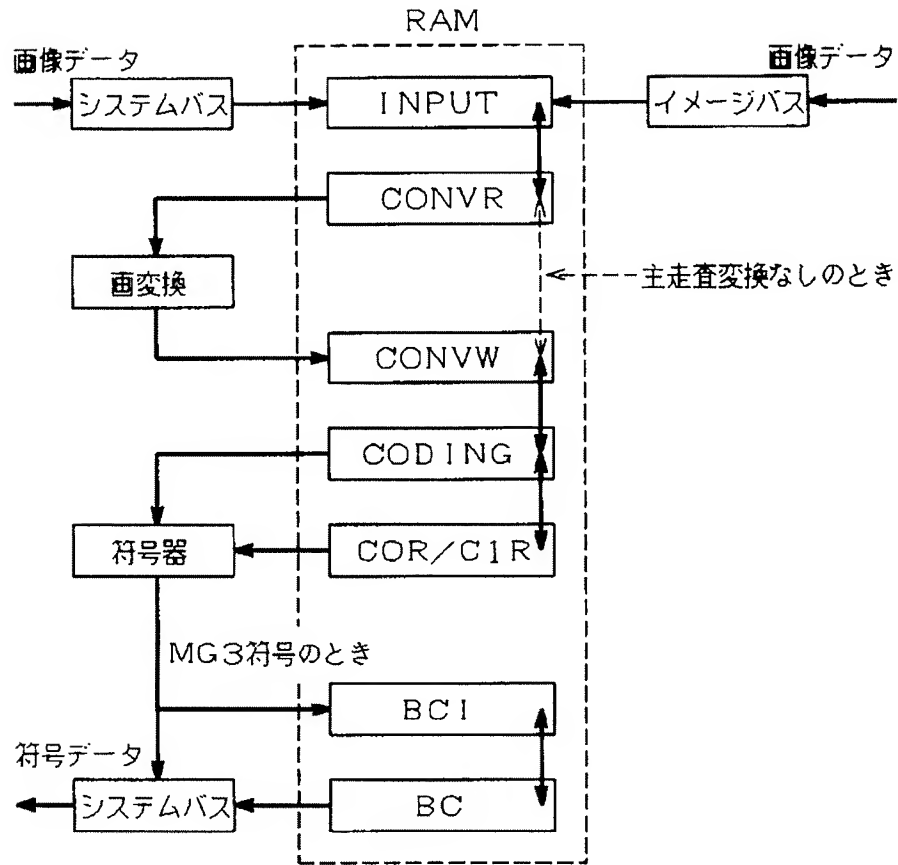
【図31】



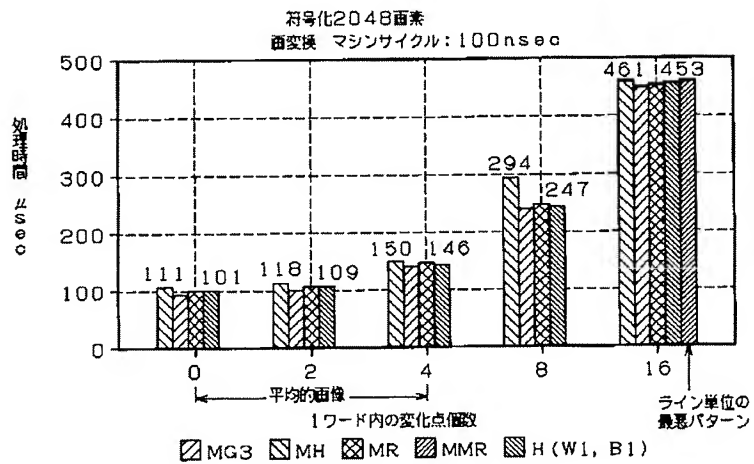
【図33】



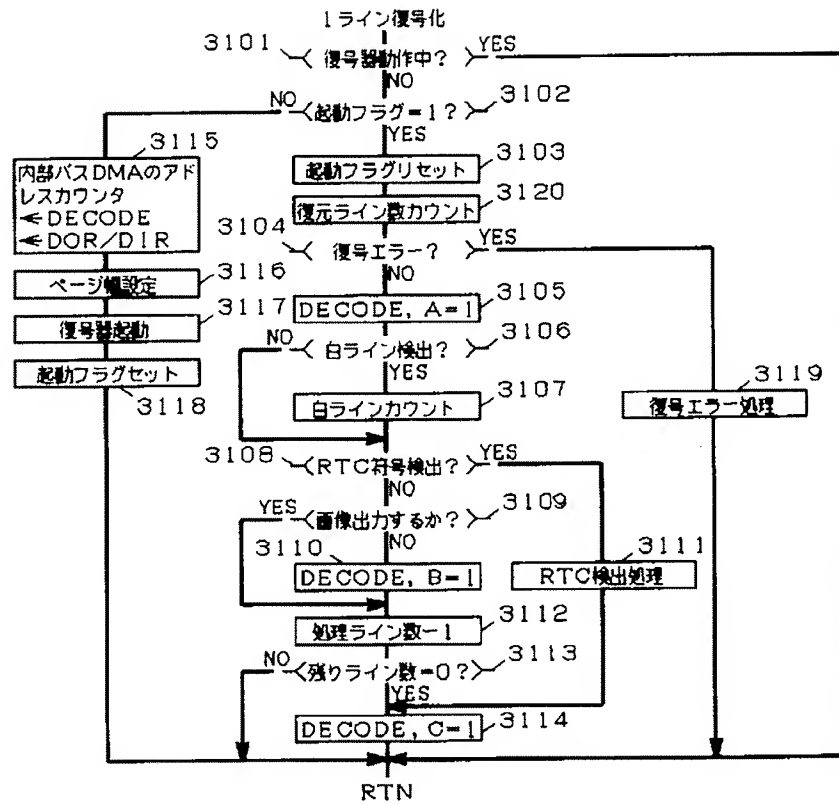
【図21】



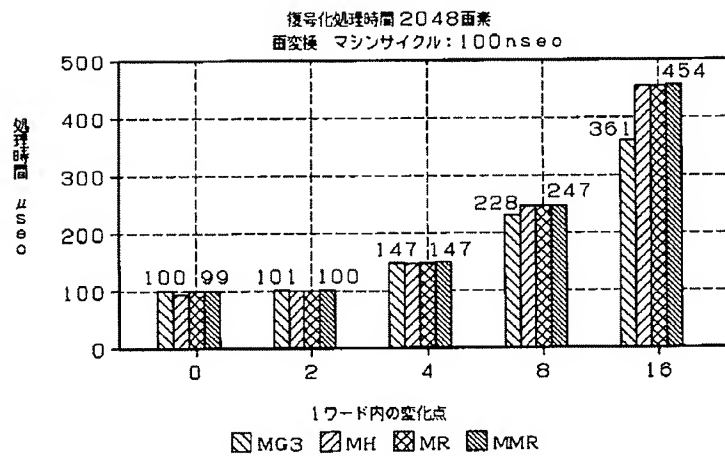
【図34】



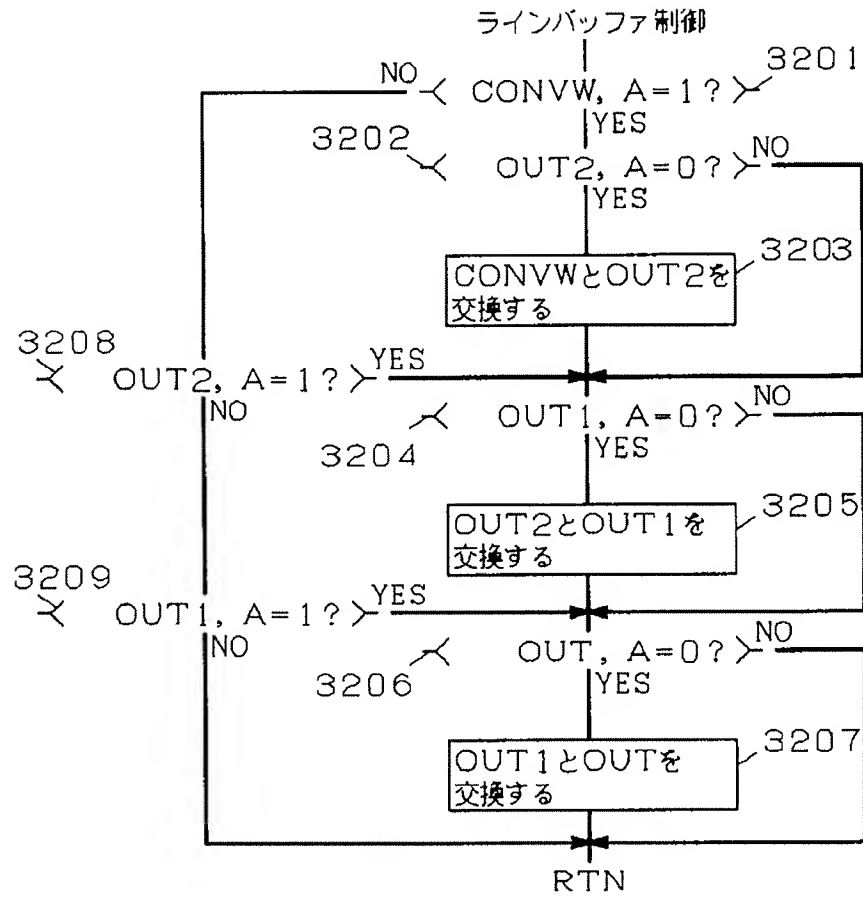
【図23】

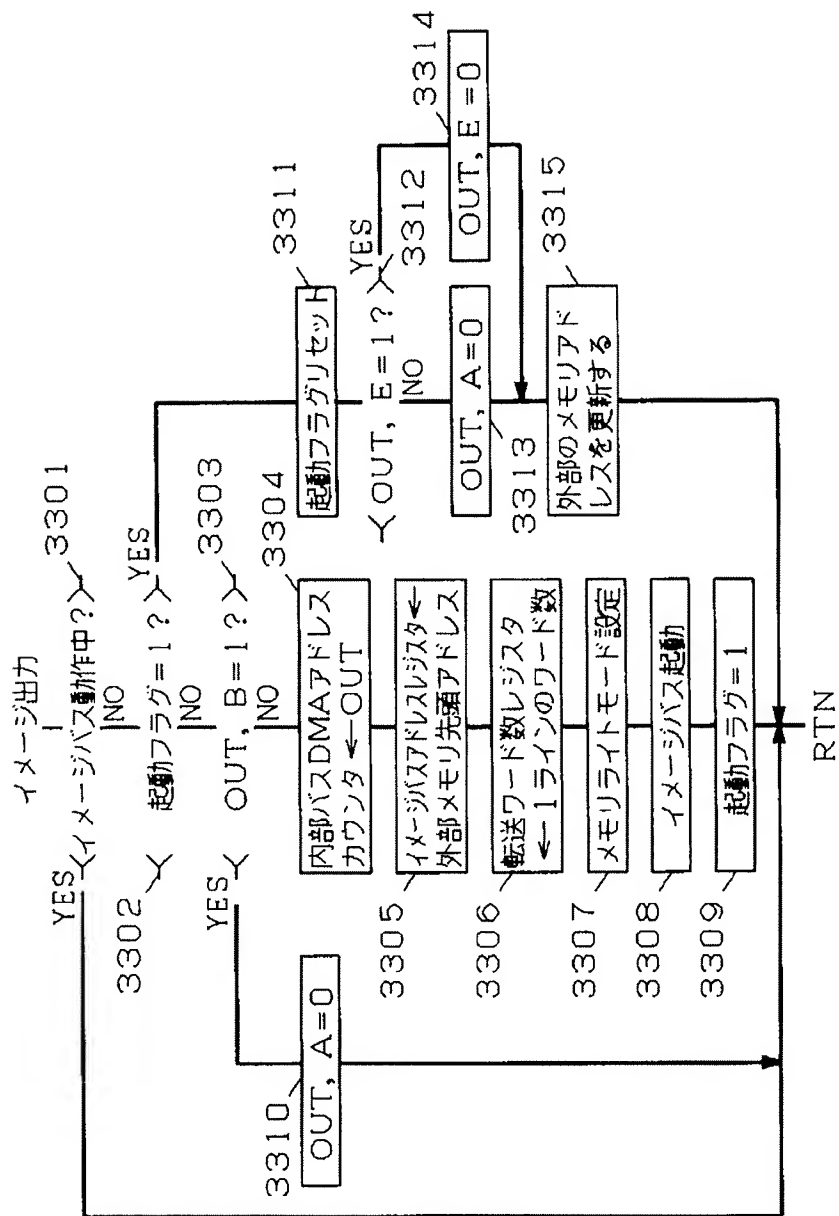


【図35】



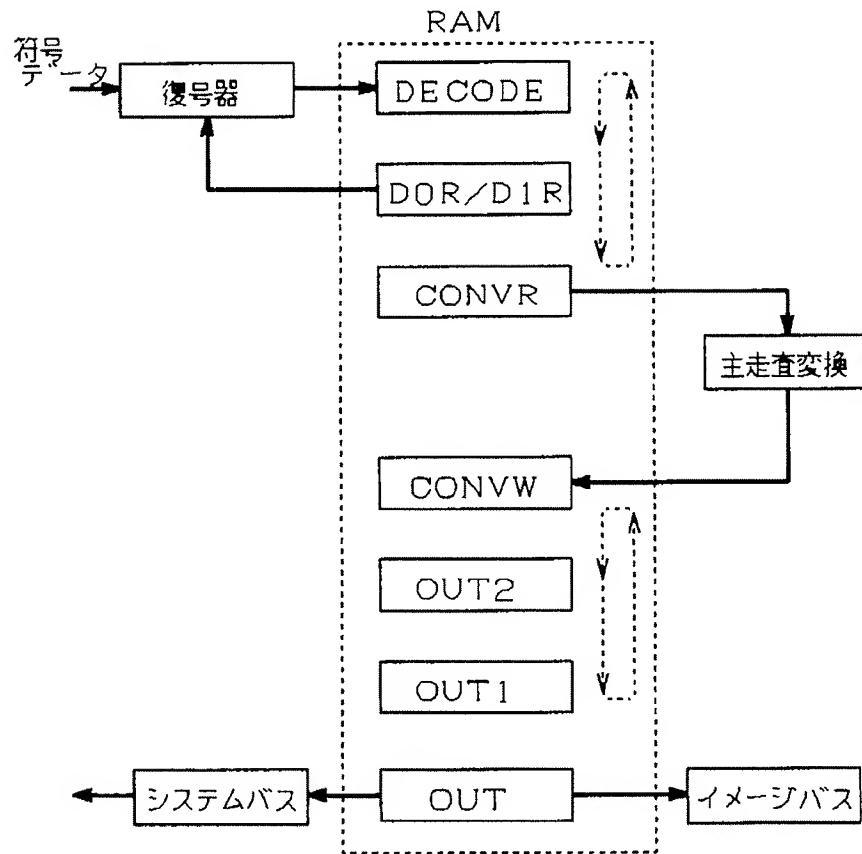
【図24】



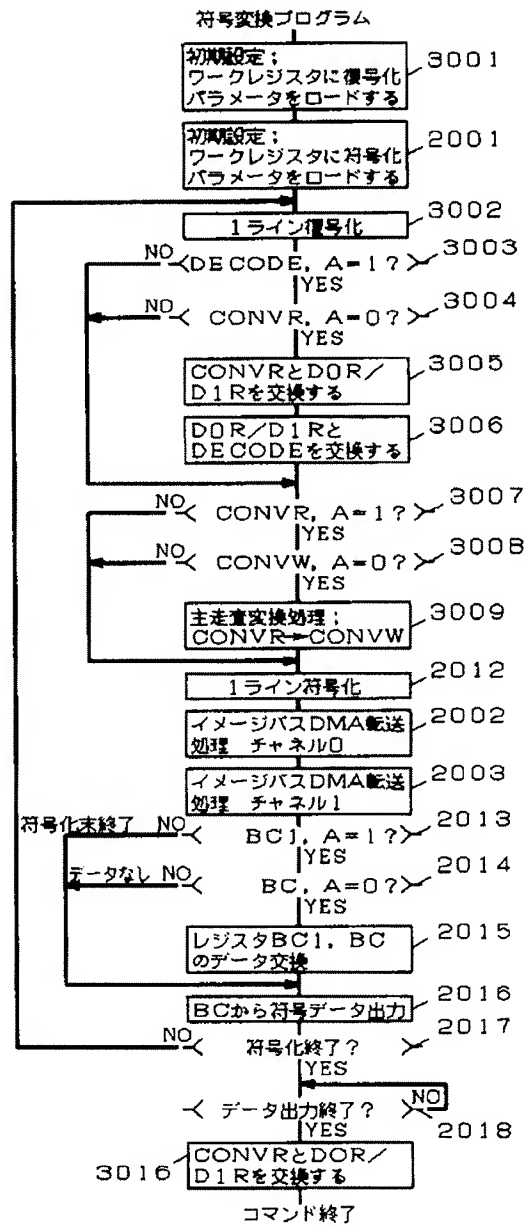


【図25】

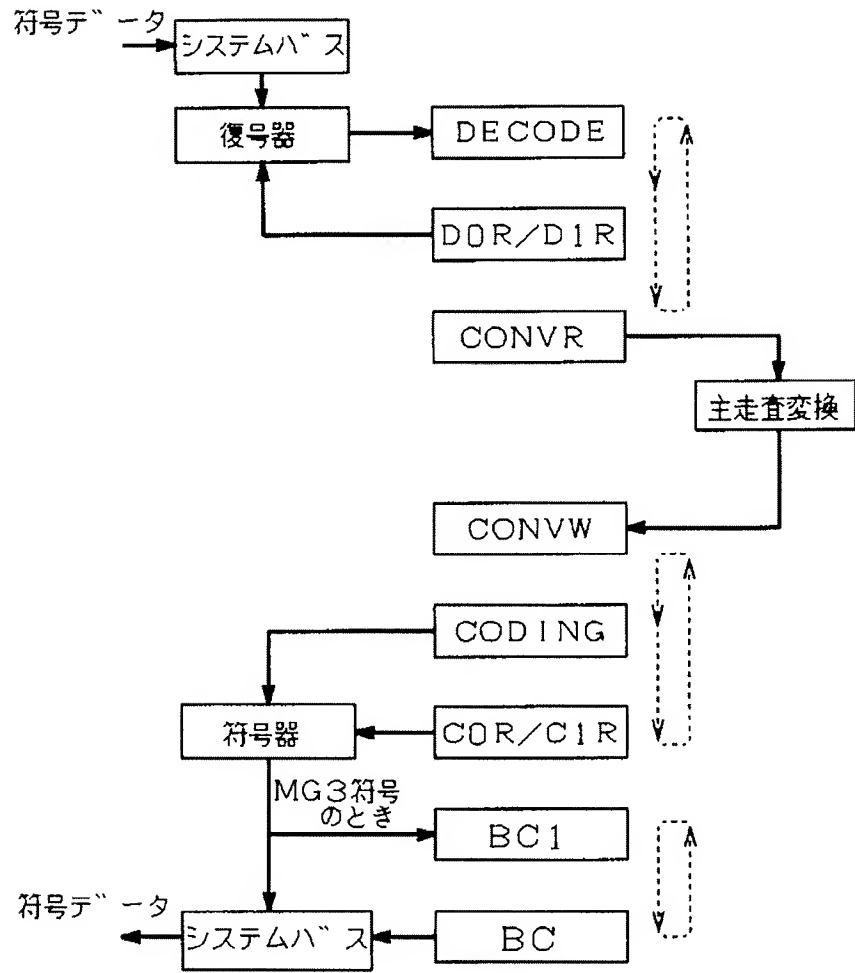
【図26】



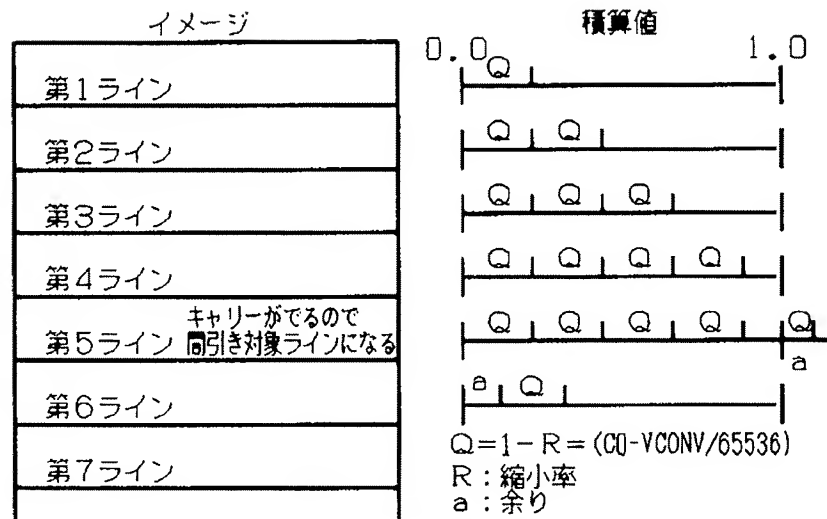
【図27】



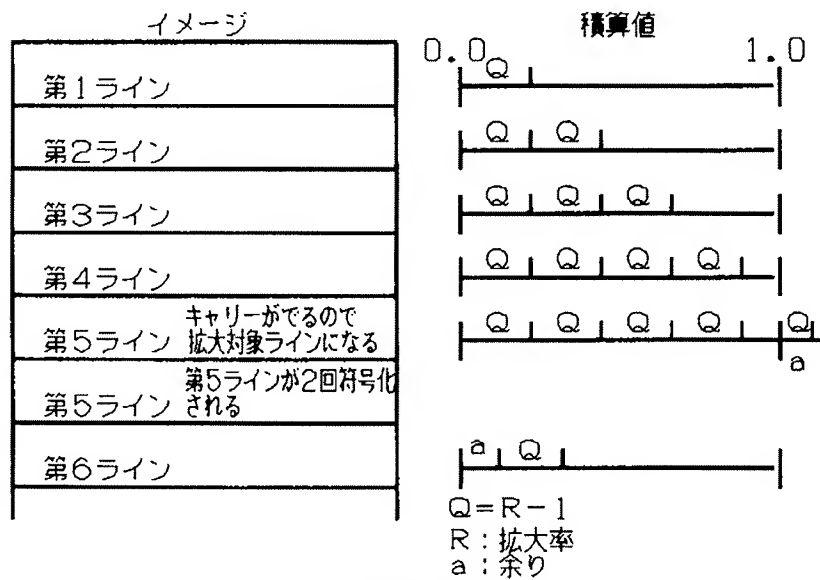
【図28】



【図30】

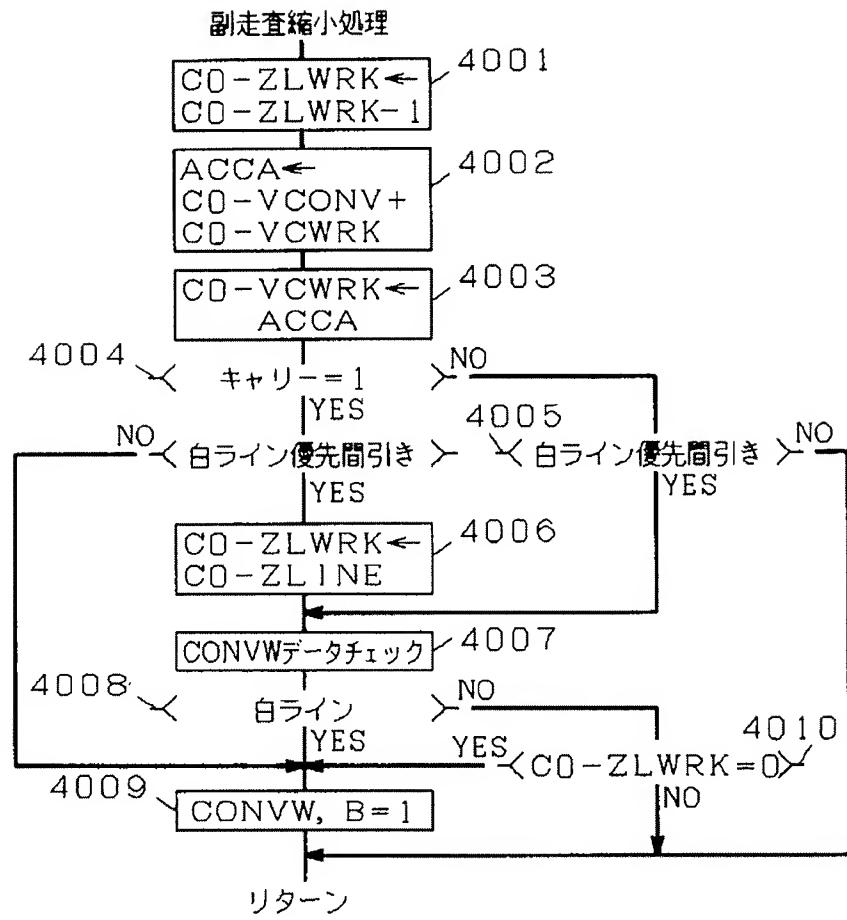


(a) 副走査方向の縮小

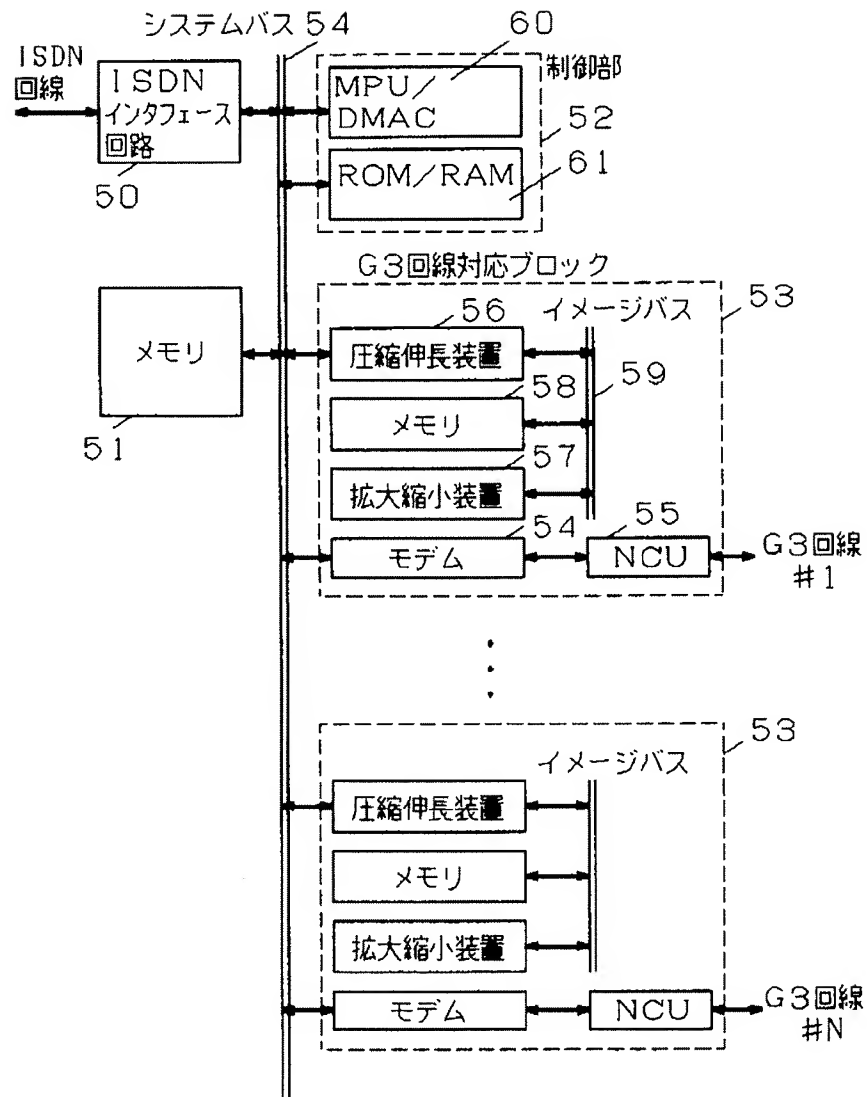


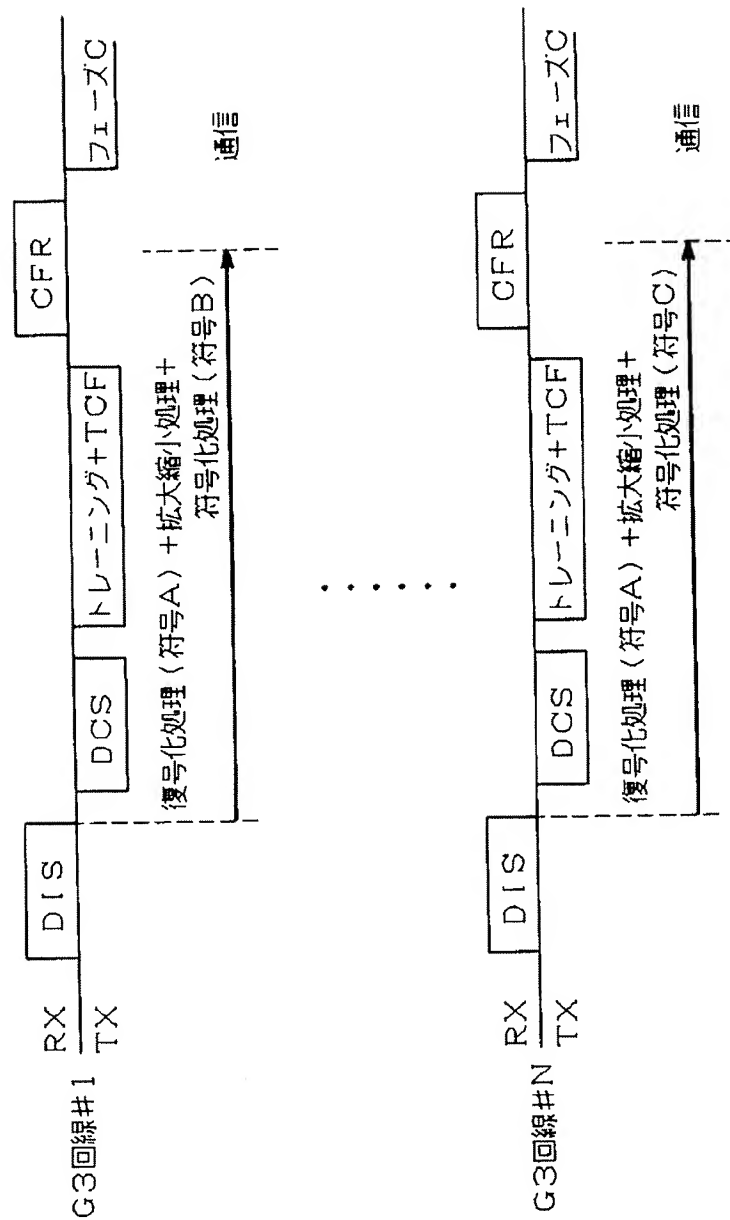
(b) 副走査方向の拡大

【図32】



【図36】





【図37】